

Research Article

The Single-Server Queue with the Dropping Function and Infinite Buffer

Andrzej Chydzinski , Marek Barczyk, and Dominik Samociuk

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

Correspondence should be addressed to Andrzej Chydzinski; andrzej.chydzinski@polsl.pl

Received 9 August 2018; Accepted 13 September 2018; Published 9 October 2018

Academic Editor: Frederico R. B. Cruz

Copyright © 2018 Andrzej Chydzinski et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present an analysis of queues with the dropping function and infinite buffer. In such queues, the arriving packet (job, customer, etc.) can be dropped with the probability which is a function of the queue size. Currently, the main application area of the dropping function is active queue management in routers, but it is applicable also in many other queueing systems. So far, queues with the dropping function have been analyzed with finite buffers only, which led to complicated, computationally demanding formulas. Assuming infinite buffers enabled us herein to obtain formulas in compact, easy to use forms. Moreover, a model with the infinite buffer can often be used as a good approximation of the real queue, in which the buffer is large. We start with noticing that the classic stability condition, $\rho < 1$, cannot be used for queues with the dropping function and infinite buffer. For this reason, we prove a few new, easy to use conditions, which guarantee system stability or instability. Then we prove several theorems on popular performance characteristics, including the queue size, busy period, loss ratio, output rate, and system response time. Additionally, we derive a special, very important characteristic called the burst ratio, which may influence severely the quality of real-time multimedia transmissions. All the theorems are illustrated with numerical examples, demonstrating in particular how the system stability may be tested and how the shape of the dropping function may affect different performance characteristics.

1. Introduction

The bufferbloat phenomenon, occurring in the contemporary Internet, consists in frequent overfilling of one or more buffers on the packet delivery path. It may cause significant performance deterioration of the network, especially in terms of the packet delivery time. This phenomenon has been diagnosed and described in several papers, (e.g., [1–3]), and the known remedy for it is an application of active queue management in Internet routers. This has been stated in the RFC 7567 document, from 2015, which begins with the following statement: “It presents a strong recommendation for testing, standardization, and widespread deployment of active queue management (AQM) in network devices to improve the performance of today’s Internet”.

Active queue management is based on preventive packet dropping, before the queue gets very long. Usually, an arriving packet is dropped with some probability, which is recomputed frequently using current or past system characteristics.

There are several possible choices of characteristics that should be used to determine the dropping probability. One of the most popular choices is computing the dropping probability, $d(n)$, as a function of the queue size, n . Function $d(n)$ is called the dropping function.

In early propositions of the dropping-function-based AQMs, a fixed shape of $d(n)$ was used in each dropping algorithm proposition. For instance, the linear dropping function was used in [4], the doubly linear one was used in [5], the exponential function was used in [6], the quadratic function was used in [7], the orthogonal polynomials were used in [8], and the cubic function was used in [9]. The performance of those algorithms was studied mainly using discrete-event simulations.

The analysis based on the queueing theory research on queues with the dropping function was initiated with approximate solutions given in [10, 11] and followed by exact results [12–19], obtained for various traffic and service assumptions. In particular, in [10], an approximate analysis

of the model with batch Poisson arrivals and linear dropping function was presented. In [11], an approximate solution of the model with general batch arrivals and exponential service was given. Exact solutions of models with single and batch Poisson arrivals and exponential services were given in [12]. The system with Poisson arrivals and general distribution of the service time was solved in [13] in the steady state, while in [14] in transient case. In [15], the model with multiple service stations was studied. In [16, 17], systems with autocorrelated arrivals were considered, with and without batch arrivals, respectively. Finally, [18, 19] deal with a generalization of the model to the case with continuous packet volumes, continuous buffer capacity, and an acceptance function (rather than dropping function), which enqueues an arriving packet with probability that depends on the remaining buffer capacity.

Now, it has to be stressed that each model in the aforementioned papers has the finite buffer. The finite-buffer assumption seems reasonable; indeed, buffers in real switches and routers are finite. Unfortunately, analytical solutions of queueing models with the dropping function and the finite buffer are rather complex (see, e.g., [17]) and computationally demanding (see, e.g., [20] for a discussion of this aspect). The reason of this is that the finite-buffer model does not enable simplification of some large, but finite sums appearing in its solution. On the other hand, similar but infinite sums in the infinite-buffer model can be often simplified to compact formulas. This situation can be observed also in the classic queueing theory (i.e., without the dropping function); solutions of infinite-buffer models are simpler, than the solutions of their finite-buffer counterparts.

To the best of the authors knowledge, there are no published results on queueing systems with the dropping function and the infinite buffer. The purpose of this paper is to perform the analysis of such queueing systems.

There are at least two reasons to study analytically such queues. The first reason was already signalled above; we can expect results in a much simpler form, than in the infinite-buffer case. Indeed, as we will see in the paper, the obtained results are more compact and easy to use. The second reason is that, given the large buffer sizes in modern devices [1], the infinite-buffer queue constitutes often a good approximation of the real queue.

There is, however, an additional cost connected with analysis of infinite-buffer models. Namely, before computing the performance characteristics, the stability of the system has to be determined. (This is not needed in the case of the finite-buffer systems; they are always stable, as the queue size is limited and cannot grow to infinity.)

There is a simple, well-known condition sufficient and necessary for stability of classic queueing systems, i.e., the system load smaller than 1 ($\rho < 1$). It is a very simple matter to check; that this condition is not necessary for stability of queues with the dropping function. For instance, a system with $\rho = 2$ and dropping function $d(n) \equiv 0.7$ is obviously stable (the arrival stream is in fact thinned to 30 percent of its original intensity). But such examples are trivial only when the dropping function is constant. For nonconstant dropping functions, we have to develop and prove new conditions on system stability. This is done in Section 3, where several new

stability tests are developed. They enable us to decide quickly whether, for instance, the system with $\rho = 1$ and $d(n) = 1/(n + 2)$ is stable, or not. (We encourage the reader to guess the stability of this system, before checking the correct answer in Section 3.)

The analysis of classic queues began with Markovian queues, i.e., with Poisson arrivals and exponential service times. We also use Markovian queues as a starting point in the analysis of queues with the dropping function and infinite buffer. It is possible that the presented methods and results can be then adjusted to non-Markovian queues, exactly as it happened in the classic theory. In all the recent studies, the dropping function does not have a fixed shape; any form of this function can be used. As shown in, e.g., [16], this allows us to control precisely several performance characteristics of interest. Therefore, the dropping function is not further specified in this paper; it can assume any form.

The rest of the paper is structured as follows. Section 2 describes the queueing model analyzed in the paper. In Section 3, theorems on the system stability are proven and accompanied with several examples of stability analysis. In Section 4, theorems on the distribution of the queue length and the average busy period are proven. In Section 5, the packet loss ratio and the output rate are derived. In Section 6, theorems on the distribution of the response time are proven in two cases: for accepted packets only and for all incoming packets. In Section 7, the burst ratio characteristic is recalled and computed for the system of interest. In Section 8, numerical examples of the mentioned characteristics are presented for a few different dropping functions. Finally, Section 9 concludes the paper.

2. Model Description

We deal with the queueing model with Poisson arrivals of intensity λ , exponential service time with average value μ , infinite waiting room (buffer), and FIFO (first in, first out) service discipline. In addition to those assumptions, we assume that every arriving packet can be dropped with probability $d(n)$, where n is the queue size at time of this packet arrival. Of course, we have to assume that $0 \leq d(n) \leq 1$ for every n . Besides that, function $d(n)$ is not further specified; it may assume any form.

The load of the system is defined traditionally as $\rho = \lambda/\mu$. By $X(t)$ we denote the queue size at time t , including the service position, if occupied. This convention will be used consequently throughout the paper; any time we say “queue size”, we mean the total number of packets in the system, including the one being serviced.

3. Stability of the System

We say that the queueing system is stable if there exist a proper limiting distribution of the queue size; i.e., for every $n = 0, 1, \dots$, there exists the limit:

$$\lim_{t \rightarrow \infty} \mathbf{P}(X(t) = n) = p_n, \quad (1)$$

and

$$\sum_{n=0}^{\infty} p_n = 1. \quad (2)$$

Otherwise, we say that the system is unstable. In the theory of Markov chains, such stable process of the queue size is called also ergodic.

As shown in many handbooks, classic queueing systems (i.e., without the dropping function) are stable if and only if $\rho < 1$. But, as argued in Section 1, we need different conditions for systems with the dropping function. We start with the basic theorem on stability.

Theorem 1. *The system with dropping function $d(n)$ and load ρ is stable, if and only if*

$$S = \sum_{k=1}^{\infty} \rho^k (1 - d(0))(1 - d(1)) \dots (1 - d(k-1)) < \infty. \quad (3)$$

Proof. Assume that, at some time t_0 , the queue size is n and that the service process is suspended. In this case, the time to the first accepted arrival has the exponential distribution with density:

$$f_n(t) = (1 - d(n)) \lambda e^{-(1-d(n))\lambda t}. \quad (4)$$

This is easy to understand, if we notice that, until the first packet is accepted, the arrival process is equivalent to the thinned Poisson process with intensity λ and thinning probability $d(n)$. It is known (see, e.g., [21]) that such thinned Poisson process is also Poisson, with rate $(1 - d(n))\lambda$. The first event in such thinned process happens after an exponentially distributed time with density $f_n(t)$, so it also must be the distribution of the first time of accepted arrival, if the service is suspended.

Now, assume that at some time t_0 , the queue size is $n > 0$, and the service is performed as usual. From the previous considerations, it follows that the queue size process either jumps down after an exponential service time with parameter μ or jumps up after an exponential service time with parameter $(1 - d(n))\lambda$, depending what happens first. If the queue size is $n = 0$ at time t_0 , then the initial arrival parameter is $(1 - d(0))\lambda$, and only a jump up is possible after an exponential service time with this parameter. After the first jump, the queue size can be either $n + 1$ or $n - 1$ and the evolution continues in the same manner.

The presented description of the queue size process is equivalent to the constructive definition of the continuous-time Markov chain, with the following intensity matrix:

$$Q = [q_{i,j}]_{i=0,1,\dots,j=0,1,\dots}, \quad (5)$$

where

$$q_{i,j} = 0, \quad i > j + 1 \text{ or } j > i + 1, \quad (6)$$

$$q_{i,i+1} = (1 - d(i)) \lambda, \quad i = 0, 1, \dots, \quad (7)$$

$$q_{i+1,i} = \mu, \quad i = 0, 1, \dots, \quad (8)$$

$$q_{i,i} = -(1 - d(i)) \lambda - \mu, \quad i = 1, 2, \dots, \quad (9)$$

$$q_{0,0} = -(1 - d(0)) \lambda. \quad (10)$$

It is known (see, e.g., [22], page 65) that a continuous-time Markov chain with such structure fulfills (1) and (2) if and only if $\sum_{k=1}^{\infty} s_k < \infty$, where

$$s_k = \frac{q_{0,1} q_{1,2} \dots q_{k-1,k}}{q_{1,0} q_{2,1} \dots q_{k,k-1}}, \quad (11)$$

Applying (6)-(10) to (11), we obtain

$$s_k = \rho^k (1 - d(0))(1 - d(1)) \dots (1 - d(k-1)). \quad (12)$$

Therefore, we can conclude that the queueing system is stable if and only if (3) is fulfilled, which completes the proof of Theorem 1. \square

Using directly condition (3) is usually inconvenient, as it is difficult to tell directly whether the series in (3) is convergent, or not. But we can use it to prove the following, very useful stability tests.

Theorem 2 (stability tests). *If at least one of conditions (a)-(e) is fulfilled,*

(a)

$$\lim_{n \rightarrow \infty} \rho [1 - d(n)] = a < 1, \quad (13)$$

(b)

$$\lim_{n \rightarrow \infty} \rho \prod_{i=1}^n \sqrt[n]{1 - d(i-1)} = a < 1, \quad (14)$$

(c)

$$\sum_{n=1}^{\infty} 2^n \rho^{2^n} \prod_{i=1}^{2^n} (1 - d(i-1)) < \infty, \quad (15)$$

(d)

$$\lim_{n \rightarrow \infty} n \frac{1 - \rho(1 - d(n))}{\rho(1 - d(n))} = a > 1, \quad (16)$$

(e) *there exists dropping function $d_0(n)$ such that $d_0(n) \leq d(n)$ for every n , and the system with d_0 and the same ρ is stable,*

then the system is stable. If at least one of conditions (f)-(j) is fulfilled,

(f) $d(n) < 1$ for every n and

$$\lim_{n \rightarrow \infty} \rho [1 - d(n)] = a > 1, \quad (17)$$

(g)

$$\lim_{n \rightarrow \infty} \rho \prod_{i=1}^n \sqrt[n]{1 - d(i-1)} = a > 1, \quad (18)$$

(h)

$$\sum_{n=1}^{\infty} 2^n \rho^{2^n} \prod_{i=1}^{2^n} (1 - d(i-1)) = \infty, \quad (19)$$

(i) $d(n) < 1$ for every n and

$$\lim_{n \rightarrow \infty} n \frac{1 - \rho(1 - d(n))}{\rho(1 - d(n))} = a < 1, \quad (20)$$

(j) there exists dropping function $d_0(n)$ such that $d_0(n) \geq d(n)$ for every n , and the system with d_0 and the same ρ is unstable,

then the system is unstable.

Proof. To show (a) and (f), assume first that $d(n) < 1$ for every n . In this case we have

$$\left| \frac{s_{n+1}}{s_n} \right| = \rho [1 - d(n)], \quad (21)$$

so points (a) and (f) follow from (3) and the ratio test for the convergence of infinite series.

If $d(n_0) = 1$ for some n_0 , then the system is stable, no matter what the value of ρ is. This is a direct consequence of (3), as in this case we have $s_n = 0$ for every $n > n_0$. Therefore, (a) is valid no matter if $d(n) < 1$ for all n , or not.

Points (b) and (g) follow from (3) and the root test for the convergence of infinite series, due to the fact that

$$\sqrt[n]{|s_n|} = \rho \prod_{i=1}^n \sqrt[n]{1 - d(i-1)}. \quad (22)$$

Applying the relation

$$2^n s_{2^n} = 2^n \rho^{2^n} \prod_{i=1}^{2^n} (1 - d(i-1)), \quad (23)$$

the Cauchy condensation test for the convergence of infinite series, and (3), we obtain points (c) and (h).

To obtain tests (d) and (i), we assume first that $d(n) < 1$ for every n , then apply (3) and the Raabe-Duhamel's test (see, e.g., [23], page 120), by computing the limit of the sequence:

$$n \left(\frac{s_n}{s_{n+1}} - 1 \right) = n \frac{1 - \rho(1 - d(n))}{\rho(1 - d(n))}. \quad (24)$$

As already shown, if $d(n_0) = 1$ for some n_0 , then the system is stable. Therefore, (d) is valid no matter if $d(n) < 1$ for all n , or not.

Point (e) follows directly from (3). Namely, if $d_0(n) \leq d(n)$ for every n , then also

$$\begin{aligned} & \rho^n (1 - d(0)) (1 - d(1)) \dots (1 - d(n-1)) \\ & < \rho^n (1 - d_0(0)) (1 - d_0(1)) \dots (1 - d_0(n-1)) \end{aligned} \quad (25)$$

for every n . Therefore, if the latter series is convergent, the former also must be convergent. In exactly the same way we can prove (j), which completes the proof of Theorem 2. \square

Remark 3. Using the generalized version of the ratio test, we can replace symbol "lim" in tests (a) and (f) with "lim sup" and "lim inf", respectively. Similarly, using the generalized version of the root test, we can replace symbol "lim" in tests (b) and (g) with "lim sup".

3.1. Stability: Examples. In the following examples, it will be demonstrated how the proposed stability tests can be used. The dropping functions in these examples are not chosen to be practically useful; they are rather meant to show the difficulties in settling whether the system is stable, or not, and demonstrate the usefulness of different tests. More practical, monotonic dropping functions will be used in numerical examples in Section 8.

(1) Let $\rho = 1$ and $d(n) = 1/(n+2)$. One may expect this system to be stable (the system with $\rho = 1$ and without the dropping function is (barely) unstable, but a strong, slowly decreasing dropping function is additionally applied, with an infinite sum of dropping probabilities, $\sum_{n=0}^{\infty} d(n) = \infty$). Surprisingly, the system is still unstable, even after application of the dropping function. This follows from test (h). We have

$$\begin{aligned} \sum_{n=1}^{\infty} 2^n \rho^{2^n} \prod_{i=1}^{2^n} (1 - d(i-1)) &= \sum_{n=1}^{\infty} 2^n \prod_{i=1}^{2^n} \frac{i}{i+1} \\ &= \sum_{n=1}^{\infty} 2^n \frac{1}{2^n + 1} = \infty. \end{aligned} \quad (26)$$

(2) Let $\rho = 1$ and $d(n) = 1/\sqrt{n+2}$. This system is stable. Now the dropping function decreases slowly enough to compensate $\rho = 1$. To show that, we may use test (d). We have

$$\lim_{n \rightarrow \infty} n \frac{1 - \rho(1 - d(n))}{\rho(1 - d(n))} = \lim_{n \rightarrow \infty} \frac{n}{\sqrt{n+2} - 1} > 1. \quad (27)$$

(3) Let $\rho = 2$ and $d(n) = 2^n/(2^{n+1} + 1)$. This system is unstable, which follows from test (j), with $d_0(n) \equiv 1/2$. Namely, for every n we have $d(n) \leq 1/2$. On the other hand, the system with $\rho = 2$ and constant dropping probability $1/2$ is equivalent to the classic system with $\rho = 1$ and no dropping function, which is unstable.

(4) Let $\rho = 1.5$ and $d(n) = n/(2n + \cos(n))$. This system is stable, which follows from test (a). We have

$$\lim_{n \rightarrow \infty} \rho (1 - d(n)) = 0.75 < 1. \quad (28)$$

It is easily seen that the system is in fact stable for any $\rho < 2$.

- (5) Let $\rho = 3$ and $d(n) = (n^3 - \log(n + 1))/(2n^3 + n + 1)$. This system is unstable, which follows from test (f). We have

$$\lim_{n \rightarrow \infty} \rho(1 - d(n)) = 1.5 > 1. \quad (29)$$

It is easy to verify that the system is in fact unstable for any $\rho > 2$.

- (6) Let $\rho = 1$ and $d(n) = (1 + \cos(n)/3)/\sqrt{n+2}$. This system is stable, which follows from test (e). Namely, we have $d(n) \geq d_0(n) = 2/(3\sqrt{n+2})$ for every n , and the system with $\rho = 1$ and dropping function $2/(3\sqrt{n+2})$ is stable, which can be shown in the same way, as in Example (2).

- (7) Let $\rho = 1$ and

$$d(n) = \begin{cases} 0, & \text{if } n \text{ is even,} \\ 10^{-6}, & \text{if } n \text{ is odd.} \end{cases} \quad (30)$$

The system is stable, which follows from test (b). Obviously, the exact value of the exponent is unimportant. Namely, replacing 10^{-6} with 10^c we obtain a stable system for any $c < 0$.

- (8) Let $\rho = 2$ and $d(n) = 1/2 + 1/(n^2 + 3)$. This system is unstable, which follows from test (i). We obtain

$$\begin{aligned} \lim_{n \rightarrow \infty} n \frac{1 - \rho(1 - d(n))}{\rho(1 - d(n))} &= \lim_{n \rightarrow \infty} n \frac{-2/(n^2 + 3)}{1 - 2/(n^2 + 3)} \\ &= \lim_{n \rightarrow \infty} -\frac{2n}{n^2 + 1} < 1. \end{aligned} \quad (31)$$

4. Queue Size and Busy Period

Theorem 4. *If the system is stable, then its stationary distribution of the queue size has the form*

$$\begin{aligned} p_0 &= (1 + S)^{-1}, \\ p_j &= p_0 \rho^j \prod_{i=1}^j (1 - d(i-1)), \quad j \geq 1, \end{aligned} \quad (32)$$

where S is defined in (3).

Proof. It is known (see, e.g., [22], page 55) that if for a Markov chain conditions (1) and (2) are fulfilled, then its stationary distribution can be determined by the set of equations:

$$q_{j,j} p_j + \sum_{\substack{i=0, \\ i \neq j}}^{\infty} p_i q_{i,j} = 0, \quad j \geq 0, \quad (33)$$

$$\sum_{j=0}^{\infty} p_j = 1, \quad (34)$$

and the solution of this set is unique. Taking into account (6)-(10), (33) can be rewritten as

$$-\lambda(1 - d(0)) p_0 + \mu p_1 = 0, \quad (35)$$

$$\begin{aligned} &[(1 - d(j)) \lambda + \mu] p_j + (1 - d(j-1)) \lambda p_{j-1} + \mu p_{j+1} \\ &= 0, \end{aligned} \quad (36)$$

for $j \geq 1$. From (35) we get

$$p_1 = p_0(1 - d(0)) \rho. \quad (37)$$

From (37) and (36) with $j = 1$ we obtain

$$\begin{aligned} p_2 &= \frac{1}{\mu} \{[(1 - d(1)) \lambda + \mu] p_1 - (1 - d(0)) \lambda p_0\} \\ &= p_0 \rho^2 (1 - d(0)) (1 - d(1)). \end{aligned} \quad (38)$$

It is easy to see that in the same way we can obtain arbitrary p_j , $j \geq 2$, in the form $p_j = p_0 \prod_{i=1}^j (1 - d(i-1)) \rho^i$, which completes the proof of the second part of (32).

As for the first part of (32), it can be proven using (34). We have

$$1 = \sum_{j=0}^{\infty} p_j = p_0 + \sum_{j=1}^{\infty} p_j = p_0 + p_0 S, \quad (39)$$

which completes the proof of Theorem 4. \square

Having computed p_n , $n = 0, 1, \dots$, we can compute the average queue size in the steady state, $\mathbb{E}X$:

$$\mathbb{E}X = \sum_{n=0}^{\infty} n p_n. \quad (40)$$

Using p_0 , we can obtain the average length of the busy period of the system, $\mathbb{E}b$.

Theorem 5. *If the system is stable, then its average duration of the busy period is equal to*

$$\mathbb{E}b = \frac{S}{\lambda(1 - d(0))}. \quad (41)$$

Proof. Formula (41) can be shown using two observations.

Firstly, from (10) we can conclude that the Markov chain, after entering state 0, stays in this state for an exponentially distributed time with parameter $\lambda(1 - d(0))$. Thus the average duration of the idle period has to be equal to

$$\frac{1}{\lambda(1 - d(0))}. \quad (42)$$

Secondly, the ratio of the average duration of the busy period to the average duration of the idle period has to be the same, as the ratio of the nonempty queue probability, to the empty queue probability. Therefore, we have

$$\frac{\mathbb{E}b}{1/[\lambda(1 - d(0))]} = \frac{1 - p_0}{p_0}. \quad (43)$$

Then we can obtain (41) from (43) and relation $p_0 = 1/(1 + S)$. \square

5. Loss Ratio and Output Rate

For a stable system, the loss ratio, L , is defined as the long-run fraction of dropped packets. It is one of the most important characteristics of queues occurring in packet network nodes. It has been measured experimentally in TCP/IP networks (see, e.g., [24–27]) and studied analytically in classic queueing models (see, e.g., [28]).

One way to compute the loss ratio in the queue with the dropping function is by using the stationary distribution of the queue size and the dropping function. From the PASTA property, we know that the queue size distribution observed at Poisson arrival times is the same as the queue size distribution observed at arbitrary times. Therefore, for a stable system we have

$$L = \sum_{j=0}^{\infty} p_j d(j). \quad (44)$$

There is, however, a simpler way to compute the loss ratio. It is based on the following formula (see, e.g., [16]):

$$L = 1 - \frac{1 - p_0}{\rho}, \quad (45)$$

which is valid for a general class of single-server queueing systems with losses. Using (45) and (32), we obtain immediately the following result.

Theorem 6. *If the system is stable, then its loss ratio equals:*

$$L = 1 - \frac{S}{(1+S)\rho}. \quad (46)$$

Using (46), we can obtain also the average intensity of the output process (see, e.g., [29], page 199):

$$\gamma = \lambda(1-L). \quad (47)$$

Namely, if the system is stable, then the average rate of the output process is

$$\gamma = \frac{S\mu}{1+S}. \quad (48)$$

6. Response Time

The system response time is defined as the total time a packet spends in the system. In systems with losses, the response time can be defined in two ways: for the accepted packets only (i.e., excluding the dropped packets) or for all the packets (i.e., including the dropped packets with the sojourn time 0). The former response time will be denoted by R_a , while the latter by R_d .

We start with derivation of the distribution of R_a .

Theorem 7. *If the system is stable, then the response time for an accepted packet has the following distribution:*

$$\begin{aligned} \mathbf{P}(R_a < t) \\ = 1 - \frac{\rho(1+S)}{S} \sum_{k=0}^{\infty} p_k (1-d(k)) \sum_{j=0}^k \frac{1}{j!} e^{-\mu t} (\mu t)^j, \end{aligned} \quad (49)$$

and the average value

$$\mathbb{E}R_a = \frac{\rho(1+S)}{S\mu} \sum_{k=0}^{\infty} (k+1) p_k (1-d(k)), \quad (50)$$

or equivalently

$$\mathbb{E}R_a = \frac{\rho(1+S)}{S\lambda} \sum_{k=0}^{\infty} k p_k. \quad (51)$$

Proof. Firstly, we need to determine the probability that an accepted packet finds n packets already present in the queue. By definition, p_n is the probability that the queue size is n at arbitrary time. From PASTA property we know that p_n is also the probability that the queue size is n at a packet arrival epoch, but this concerns all packets, dropped and accepted. We are interested in fact in the conditional probability $\mathbf{P}(A_n | B)$, where A_n is the event that the queue size is n upon a packet arrival, while B is the event that an arriving packet is accepted. Obviously, we have $\mathbf{P}(B) = 1 - L$, while probability that upon a packet arrival the queue size is n and the packet is accepted is $p_n(1-d(n))$. Therefore, we have $\mathbf{P}(A_n | B) = p_n(1-d(n))/(1-L)$.

Secondly, the remaining service time of the packet being serviced upon arrival of a new packet is exponentially distributed with parameter μ . This follows from the memoryless property of the service time distribution.

Therefore, if just before the arrival of an accepted packet the queue size is k , this newly accepted packet will spend $k+1$ full service cycles in the system, each exponentially distributed with parameter μ . Thus the response time of such packet has Erlang distribution function, with parameters $k+1$ and μ ; i.e.,

$$E(k+1, \mu) = 1 - \sum_{j=0}^k \frac{1}{j!} e^{-\mu t} (\mu t)^j. \quad (52)$$

Summarizing these considerations, we obtain

$$\begin{aligned} \mathbf{P}(R_a < t) &= \sum_{k=0}^{\infty} \frac{p_k(1-d(k))}{1-L} E(k+1, \mu) \\ &= \sum_{k=0}^{\infty} \frac{p_k(1-d(k))}{1-L} \left(1 - \sum_{j=0}^k \frac{1}{j!} e^{-\mu t} (\mu t)^j \right), \quad (53) \\ &= 1 - \frac{1}{1-L} \sum_{k=0}^{\infty} p_k (1-d(k)) \sum_{j=0}^k \frac{1}{j!} e^{-\mu t} (\mu t)^j, \end{aligned}$$

which, taking into account (46), gives (49).

From (49) we can get (50). Namely, we have

$$\begin{aligned}
 \mathbb{E}R_a &= \int_0^{\infty} (1 - \mathbf{P}(R_a < t)) dt \\
 &= \frac{\rho(1+S)}{S} \sum_{k=0}^{\infty} p_k (1-d(k)) \sum_{j=0}^k \frac{1}{j!} \int_0^{\infty} e^{-\mu t} (\mu t)^j dt, \\
 &= \frac{\rho(1+S)}{S} \sum_{k=0}^{\infty} p_k (1-d(k)) \sum_{j=0}^k \frac{1}{\mu^j} \\
 &= \frac{\rho(1+S)}{S\mu} \sum_{k=0}^{\infty} (k+1) p_k (1-d(k)).
 \end{aligned} \tag{54}$$

To show (51), we can use Little's theorem. Applying this theorem yields

$$\mathbb{E}X = \lambda(1-L)\mathbb{E}R_a. \tag{55}$$

Thus we have

$$\mathbb{E}R_a = \frac{\mathbb{E}X}{\lambda(1-L)} = \frac{\sum_{k=0}^{\infty} k p_k}{\lambda(1-L)} = \frac{\rho(1+S)}{S\lambda} \sum_{k=0}^{\infty} k p_k, \tag{56}$$

which completes the proof of the theorem. \square

We can calculate also the distribution of R_d .

Theorem 8. *If the system is stable, then the response time for an arbitrary packet, accepted or not, has the following distribution:*

$$\mathbf{P}(R_d < t) = 1 - \sum_{k=0}^{\infty} p_k (1-d(k)) \sum_{j=0}^k \frac{1}{j!} e^{-\mu t} (\mu t)^j, \tag{57}$$

and the following average value:

$$\mathbb{E}R_d = \frac{1}{\mu} \sum_{k=0}^{\infty} (k+1) p_k (1-d(k)), \tag{58}$$

or equivalently

$$\mathbb{E}R_d = \frac{1}{\lambda} \sum_{k=0}^{\infty} k p_k. \tag{59}$$

Proof. When an arriving packet is dropped, its response time, R_d , is 0. As for an accepted packet the response time is always nonzero, we conclude that the packet loss is the only case, when the response time is 0. Therefore, we have

$$\mathbf{P}(R_d = 0) = L = 1 - \frac{S}{(1+S)\rho}. \tag{60}$$

Reasoning in the similar way, as in the proof of the previous theorem, for $t > 0$ we have

$$\begin{aligned}
 \mathbf{P}(R_d < t) &= L + \sum_{k=0}^{\infty} p_k (1-d(k)) E(k+1, \mu) \\
 &= L + \sum_{k=0}^{\infty} p_k (1-d(k)) \left(1 - \sum_{j=0}^k \frac{1}{j!} e^{-\mu t} (\mu t)^j \right) \\
 &= 1 - \sum_{k=0}^{\infty} p_k (1-d(k)) \sum_{j=0}^k \frac{1}{j!} e^{-\mu t} (\mu t)^j,
 \end{aligned} \tag{61}$$

which completes the proof of (57).

Finally, (58), (59) can be obtained from (50), (51), respectively, using the following relation:

$$\mathbb{E}R_d = L \cdot 0 + (1-L)R_a. \tag{62}$$

This completes the proof of the theorem. \square

7. Burst Ratio

As proposed in [30], the burst ratio, B , is a characteristic of tendency of packet losses to cluster in long sequences (note that the term is used also in different meanings, but herein we use it strictly in the meaning of patent [30]).

The burst ratio is defined as the average length of the sequence of consecutive packet losses, divided by the average length of the sequence of consecutive losses in the Bernoulli process (i.e., when the losses are independent on each other). As the average length of the sequence of losses in the Bernoulli process equals $1/(1-L)$, the burst ratio can be computed as

$$B = \overline{G}(1-L), \tag{63}$$

where \overline{G} is the average length of the sequence of losses in the stream of interest.

The burst ratio has a simple interpretation. Namely, $B > 1$ means that there is a tendency of losses to cluster together, $B < 1$ indicates that the losses are more spread than could be expected in the random case, while $B = 1$ is neutral.

The burst ratio in finite-buffer queueing systems without the dropping function has been studied recently in [31, 32]. Among other things, it was shown that the buffering itself is responsible for tendency of losses in networks to cluster in long sequences; in typical queueing scenarios the values of burst ratio were significantly greater than 1. Moreover, in [31] the system with single-arrival stream was considered, while in [32] the system with bursty traffic was modeled by the batch Poisson process. The bursty traffic in [32] resulted in far worse performance in terms of the average length of the sequence of consecutive packet losses. This is yet another characteristic that gets worse when the traffic is bursty. The general deterioration of queueing performance (throughput, delay, and blocking probability) with traffic burstiness has been known for a long time. For instance, in [33] the negative effects of traffic burstiness on the flow

control mechanism were shown, using finite-buffer queueing system with the doubly stochastic Poisson traffic model (with the state-dependent arrival rate).

We deal herein with B for two reasons. Firstly, B has a deep impact on real-time multimedia transmissions; long sequences of losses cause drastic deterioration of such transmissions. For instance, in [34] the following impairment factor for the packetized voice transmission is proposed, as a direct function of B and L :

$$I = I_e + (95 - I_e) \frac{100L}{100L/B + R}, \quad (64)$$

where I_e and R are some parameters, unimportant for this considerations. Secondly, it is intuitively clear that an application of the dropping function should make B smaller. This conjecture has been already confirmed by simulations in [35]. We confirm this further in Section 8, using the theorem proven below.

Theorem 9. *If the system is stable, then the packet burst ratio is equal to*

$$B = \frac{S \sum_{n=1}^{\infty} \hat{p}_n \bar{r}_n}{(1+S) \rho \sum_{n=1}^{\infty} (1-r_{n,0}) \hat{p}_n}, \quad (65)$$

where

$$\hat{p}_n = \frac{\rho(1+S)(1-d(n-1))p_{n-1}}{S}, \quad (66)$$

$$\bar{r}_n = \sum_{k=1}^{\infty} k r_{n,k}, \quad (67)$$

$$r_{0,0} = 1 - d(0), \quad (68)$$

$$r_{0,k} = d^k(0)(1-d(0)), \quad k = 1, 2, \dots, \quad (69)$$

$$r_{n,0} = \frac{\mu^n(1-d(0))}{(\lambda+\mu)^n} + \lambda \sum_{j=0}^{n-1} \frac{\mu^j(1-d(n-j))}{(\lambda+\mu)^{j+1}}, \quad (70)$$

$n = 1, 2, \dots,$

$$r_{n,k} = \frac{1}{\lambda+\mu} [\lambda d(n) r_{n,k-1} + \mu r_{n-1,k} r], \quad (71)$$

$$n = 1, 2, \dots, \quad k = 1, 2, \dots$$

Proof. Let us mark all the packets arriving to the buffer by two letters: a , if the packet is accepted, or d , if the packet is dropped. For example, the stream of arriving packets may look like this:

$$a \ a \ a \ d \ a \ d \ d \ d \ a \ d \ d \ a. \quad (72)$$

It will be useful for us to perceive this stream in such a way that every accepted packet finishes the previous sequence of dropped packets and opens a new one. In other words, between two consecutive acceptances we always have a sequence of losses, possibly of length 0. Using such convention, we can rewrite the example given above as

$$a \ 0 \ a \ 0 \ a \ 1 \ a \ 3 \ a \ 2 \ a, \quad (73)$$

where each a denotes an accepted packet and the numbers between two accepted packets denote the numbers of dropped packets. Let \bar{R} denote the average of these numbers, i.e., the average length of sequence of dropped packets (including sequences of length 0). The computation of \bar{R} will constitute the most important part of the proof.

Because the system drops packets differently, depending on the queue size, to compute \bar{R} , we have to compute first the probability \hat{p}_n , that just after a packet acceptance, the queue size is n . Using PASTA property and formula (46), we get

$$\begin{aligned} \hat{p}_n &= \frac{1}{1-L} (1-d(n-1)) p_{n-1} \\ &= \frac{\rho(1+S)(1-d(n-1))p_{n-1}}{S} \quad n = 1, 2, \dots, \end{aligned} \quad (74)$$

which completes the proof of (66).

Now, assume that, at some time t_0 , the queue size is n , and denote by $r_{n,k}$ the probability, that starting from time t_0 , exactly k arriving packets will be dropped in a row, before the first acceptance. (Note that, due to Markovian structure of the queue size process, $r_{n,k}$ does not depend on the choice of t_0 .) We will show now that probabilities $r_{n,k}$ have the form given in formulas (68)-(71).

Firstly, starting from queue size 0, the sequence of losses of length 0 happens only if the first arriving packet gets accepted. As this happens with probability $1-d(0)$, we have proven (68). Again, starting from queue size 0, the sequence of losses of length $k \geq 1$ happens only if k arriving packets are dropped and the next one is accepted. This happens with probability $d^k(0)(1-d(0))$, so we have proven (69).

The case with the initial queue size greater than 0 is more difficult. This is due to the fact that the queue size may decrease in the middle of the sequence of consecutive drops and the dropping probability is not constant, as in the previous paragraph.

To show (70), we start from $n = 1$. With probability $\lambda/(\lambda+\mu)$, the first event in the system is a packet arrival. To assure $k = 0$, this packet has to be accepted, which happens with probability $1-d(1)$. On the other hand, with probability $\mu/(\lambda+\mu)$ the first event in the system is a packet departure. In that case, the new queue size is 0, and the probability that the next packet will be accepted is $1-d(0)$. Therefore, we obtain

$$r_{1,0} = \frac{\lambda}{\lambda+\mu} (1-d(1)) + \frac{\mu}{\lambda+\mu} (1-d(0)). \quad (75)$$

For $n = 2$, we can proceed in the same manner. With probability $\lambda/(\lambda+\mu)$, the first event is an arrival; in this case, the packet has to be accepted. On the other hand, with probability $\mu/(\lambda+\mu)$, the first event is a departure; in that case, the new queue size is 1, and the probability that the next packet will be accepted is $r_{1,0}$. Thus we have

$$\begin{aligned} r_{2,0} &= \frac{\lambda}{\lambda+\mu} (1-d(2)) + \frac{\mu}{\lambda+\mu} r_{1,0} \\ &= \frac{\mu^2(1-d(0))}{(\lambda+\mu)^2} + \frac{\lambda\mu(1-d(1))}{(\lambda+\mu)^2} + \frac{\lambda(1-d(2))}{\lambda+\mu}. \end{aligned} \quad (76)$$

Proceeding by induction, we can complete the proof of (70).

We can use similar arguments, to show formula (71) for every $n, k \geq 1$. With probability $\lambda/(\lambda + \mu)$, the first event is an arrival. In this case, the first packet has to be dropped. Therefore, the queue size does not change and next $k - 1$ packets have to be dropped as well; this happens with probability $r_{n,k-1}$. On the other hand, with probability $\mu/(\lambda + \mu)$ the first event is a departure. In that case, the new queue size is $n - 1$, and the next k packets have to be dropped, which happens with probability $r_{n-1,k}$.

Having computed $r_{n,k}$ for every n, k , we can compute \bar{r}_n , i.e., the average length of the sequence of dropped packets, starting from the queue size n . It is given in (67).

Having computed \hat{p}_n, \bar{r}_n , and taking into account the fact that each accepted packet finishes one sequence of losses and begins a new one (as in (73)), we can compute \bar{R} ; namely,

$$\bar{R} = \sum_{n=1}^{\infty} \hat{p}_n \bar{r}_n. \quad (77)$$

Now, in the definition of \bar{R} , we included sequences of losses of length 0. When computing the burst ratio, such sequences have to be excluded, and the remaining probability mass has to be scaled properly to 1. The probability that after a packet acceptance the sequence of losses is not empty is equal to

$$\sum_{n=1}^{\infty} (1 - r_{n,0}) \hat{p}_n. \quad (78)$$

Therefore, the average length of nonempty sequence of losses is

$$\bar{G} = \frac{\bar{R}}{\sum_{n=1}^{\infty} (1 - r_{n,0}) \hat{p}_n} = \frac{\sum_{n=1}^{\infty} \hat{p}_n \bar{r}_n}{\sum_{n=1}^{\infty} (1 - r_{n,0}) \hat{p}_n}. \quad (79)$$

Finally, combining (63) with (79) and (46), we arrive at (65), which completes the proof of the theorem. \square

8. Numerical Examples

In these examples, we use the critically loaded system, with $\lambda = 1$ and $\mu = 1$, and the following four dropping functions:

$$\begin{aligned} d_0(n) &= \begin{cases} 0, & \text{if } n < 100, \\ 1, & \text{otherwise,} \end{cases} \\ d_1(n) &= \begin{cases} e^{(n-100)/15}, & \text{if } n < 100, \\ 1, & \text{otherwise,} \end{cases} \\ d_2(n) &= \begin{cases} 0, & \text{if } n < 100, \\ 1 - e^{-(n-100)/20}, & \text{otherwise,} \end{cases} \\ d_3(n) &= \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{n-100}{4}\right), \end{aligned} \quad (80)$$

which are depicted in Figure 1. As we can see, d_0 represents in fact the lack of the dropping function (i.e., the classic FIFO-drop-tail queueing); it does not drop any packet, when the

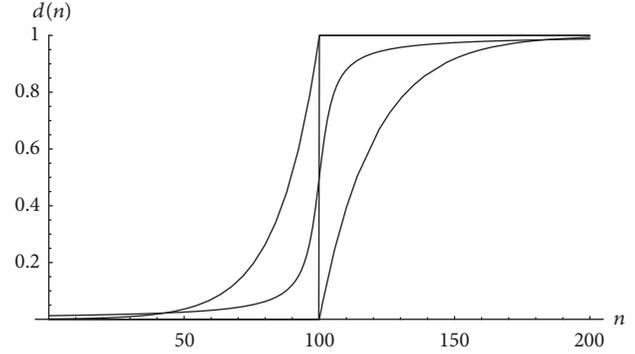


FIGURE 1: Dropping functions d_0-d_3 . Function d_1 on the left, d_2 on the right, and d_0 and d_3 in the middle.

TABLE 1: Performance characteristics for dropping functions d_0-d_3 .

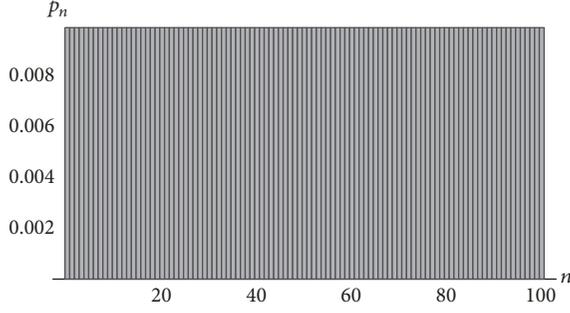
characteristic	d_0	d_1	d_2	d_3
average queue size, $\mathbb{E}X$	50.00	28.53	52.86	30.24
avg. response time, $\mathbb{E}R_a$	50.50	29.08	53.36	30.93
loss ratio, L	0.99%	1.89%	0.93%	2.24%
burst ratio, B	1.980	1.045	1.239	1.015
average busy period, $\mathbb{E}b$	100.0	51.76	105.6	44.06

queue size is below 100, and drops all packets, when the queue size reaches 100 (full buffer). Function d_1 also drops all packets, when the queue size reaches 100 packets, but the dropping probability is a nonzero, convex function on interval $(0,100)$, which initially grows rather slowly, but then rapidly. Function d_2 does not drop any packet in interval $(0,100)$, but then it grows rapidly as a concave function of n . Finally, d_3 is a convex function in interval $(0,100)$, smaller than d_1 , and concave in interval $(100,\infty)$, greater than d_2 . All functions d_0-d_3 are monotonic and nondecreasing. Nondecreasing dropping functions are most popular in networking, because there is no practical reason to decrease the dropping probability when the queue size grows. There is no agreement on, however, which shape of the dropping function is the best. We will show below that this is due to the fact that there is no obvious way to define what “the best” means in the context of the dropping function.

Firstly, we can easily check, using test (a) of Theorem 2, that the system is stable for $\rho = 1$ and each of the mentioned dropping functions. Therefore, we can compute all their performance characteristics, derived in the previous sections. The results are gathered in Table 1 and Figures 2–5. In the networking context, the most important characteristics are the first four in Table 1; the busy period is not easily translated into the network performance.

In Table 1 and Figures 2–5, we can compare the performance of “proper” dropping functions, d_1-d_3 , with each other, as well as with the lack of the dropping function (d_0 case).

Comparing results for functions d_1-d_3 with each other, we see that each function has the best performance in some area. Namely, d_1 provides the shortest queue size and response time on average; d_2 gives an excellent loss ratio, while d_3

FIGURE 2: Stationary distribution of the queue size for d_0 .TABLE 2: Theoretical versus simulation results for d_2 .

characteristic	theory	simulation
average queue size, $\mathbb{E}X$	52.8600	52.8586
avg. response time, $\mathbb{E}R_a$	53.3604	53.3612
loss ratio, L	0.93773%	0.93860%
burst ratio, B	1.23971	1.24026
average busy period, $\mathbb{E}b$	105.640	105.497

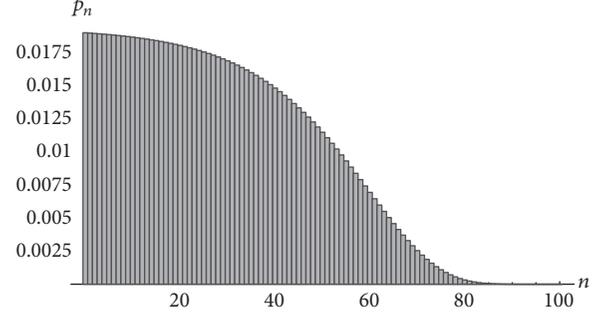
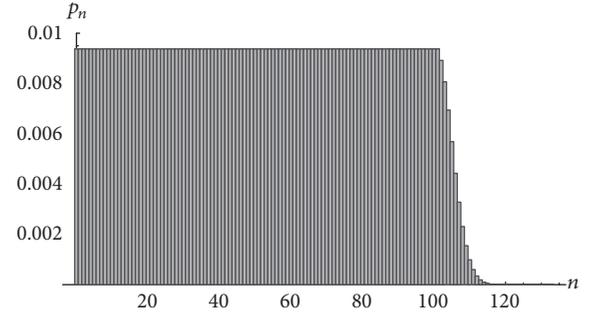
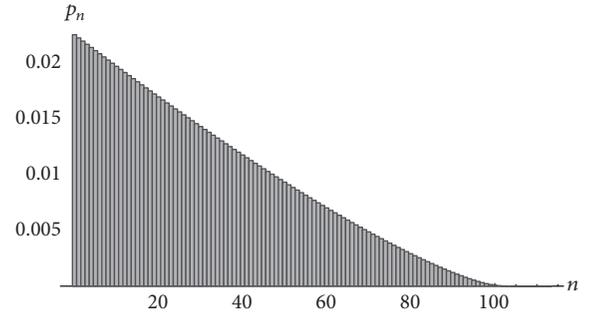
TABLE 3: Theoretical versus simulation results for d_3 .

characteristic	theory	simulation
average queue size, $\mathbb{E}X$	30.2437	30.2010
avg. response time, $\mathbb{E}R_a$	30.9389	30.8957
loss ratio, L	2.24694%	2.24394%
burst ratio, B	1.01533	1.01530
average busy period, $\mathbb{E}b$	44.0656	44.0395

provides a superb burst ratio, very close to 1 (and not a bad queue size as well). At the same time, each of functions d_1 - d_3 has a weakness. In the case of d_1 and d_3 , it is a higher loss ratio, while in the case of d_2 , it is a higher queue size. Therefore, it is impossible to tell which one is the best in general. To do this, one should propose first and justify properly, a cost function, which would combine at least characteristics $\mathbb{E}X$, L , and B in one formula.

Comparing the performance of functions d_1 - d_3 with the case of no proper dropping function, d_0 , we can say roughly that an application of the dropping function is generally profitable. For instance, comparing the results for d_2 with those for d_0 , we see that d_2 reduces the loss ratio by about 6% and reduces the burst ratio by about 37%, at the cost of the queue size increased by about 5%. Especially good results can be observed in the reduction of the burst ratio, for all considered functions d_1 - d_3 .

Finally, we checked the correctness of the proven theoretical results by performing simulations in Omnet++ environment, [36]. The results obtained for 5×10^8 packets passing through the queueing system and dropping functions d_2 , d_3 are gathered in the third column of Tables 2 and 3, respectively. For comparison, the results obtained via theorems are given in the second column. We can observe a high agreement between the theoretical and simulation results in all cases.

FIGURE 3: Stationary distribution of the queue size for d_1 .FIGURE 4: Stationary distribution of the queue size for d_2 .FIGURE 5: Stationary distribution of the queue size for d_3 .

9. Conclusions

We presented an analysis of queueing systems with the dropping function and infinite buffer. In particular, we gave several conditions to test the system stability and demonstrated them on examples. We derived all the most important performance characteristics: the queue size distribution, the busy period, the loss ratio, the system response time, and the burst ratio. We illustrated the theorems with numerical examples, which demonstrated how profitable an application of the dropping function can be and how the shape of the dropping function may affect different performance characteristics.

One of possible directions of future work was signalled in Section 8. Namely, the dropping function can be used to improve several performance characteristics of the queueing system, at the cost of some degradation of other characteristics. To fully exploit the potential of the dropping function, we

need an overall performance cost function, combining all the most important performance characteristics in one formula. Such overall cost function could be then optimized using different forms of the dropping function.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was conducted within Project 2017/25/B/ST6/00110, funded by National Science Centre, Poland.

References

- [1] J. Gettys and K. Nichols, "Bufferbloat: Dark Buffers in the Internet," *Queue*, vol. 9, no. 11, pp. 1–15, 2011.
- [2] K. Nichols and V. Jacobson, "Controlling queue delay," *Communications of the ACM*, vol. 55, no. 7, pp. 42–50, 2012.
- [3] V. G. Cerf, "Bufferbloat and other internet challenges," *IEEE Internet Computing*, vol. 18, no. 5, pp. 79–80, 2014.
- [4] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.
- [5] V. Rosolen, O. Bonaventure, and G. Leduc, "A RED discard strategy for ATM networks and its performance evaluation with TCP/IP traffic," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 3, pp. 23–43, 1999.
- [6] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, 2001.
- [7] K. Zhou, K. L. Yeung, and V. Li, "Nonlinear RED: a simple yet efficient active queue management scheme," *Computer Networks*, vol. 50, no. 18, pp. 3784–3794, 2006.
- [8] D. R. Augustyn, A. Domański, and J. Domańska, "A Choice of Optimal Packet Dropping Function for Active Queue Management," *Communications in Computer and Information Science*, vol. 79, pp. 199–206, 2010.
- [9] J. Domanska, D. R. Augustyn, and A. Domanski, "The choice of optimal 3-rd order polynomial packet dropping function for NLRED in the presence of self-similar traffic," *Bulletin of the Polish Academy of Sciences—Technical Sciences*, vol. 60, no. 4, pp. 779–786, 2012.
- [10] T. Bonald, M. May, and J.-C. Bolot, "Analytic evaluation of RED performance," in *Proceedings of INFOCOM*, pp. 1415–1424, 2000.
- [11] H. Wang and W. Yan, "An extended $G^X/M/1/N$ queueing model for evaluating the performance of AQM algorithms with aggregate traffic," *Lecture Notes in Computer Science*, vol. 3619, pp. 395–414, 2005.
- [12] W. M. Kempa, "On main characteristics of the $M/M/1/N$ queue with single and batch arrivals and the queue size controlled by AQM algorithms," *Kybernetika*, vol. 47, no. 6, pp. 930–943, 2011.
- [13] A. Chydzinski and L. Chrost, "Analysis of AQM queues with queue-size based packet dropping," *Applied Mathematics and Computer Science*, vol. 21, no. 3, pp. 567–577, 2011.
- [14] W. M. Kempa, "A direct approach to transient Queue-Size distribution in a finite-buffer Queue with AQM," *Applied Mathematics & Information Sciences*, vol. 7, no. 3, pp. 909–905, 2013.
- [15] O. Tikhonenko and W. M. Kempa, "Performance evaluation of an $M/G/N$ -type queue with bounded capacity and packet dropping," *Applied Mathematics and Computer Science*, vol. 26, no. 4, pp. 841–854, 2016.
- [16] A. Chydzinski and P. Mrozowski, "Queues with Dropping Functions and General Arrival Processes," *PLoS ONE*, vol. 11, no. 3, pp. 1–23, 2016.
- [17] P. Mrozowski and A. Chydzinski, "Queues with dropping functions and autocorrelated arrivals," *Methodology and Computing in Applied Probability*, vol. 20, no. 1, pp. 97–115, 2018.
- [18] O. Tikhonenko and W. M. Kempa, "The generalization of AQM algorithms for queueing systems with bounded capacity," *Lecture Notes in Computer Science*, vol. 7204, pp. 242–251, 2012.
- [19] O. Tikhonenko and W. M. Kempa, "Erlang service system with limited memory space under control of AQM mechanism," *Communications in Computer and Information Science*, vol. 718, pp. 366–379, 2017.
- [20] P. Mrozowski and A. Chydzinski, "Matrix inversion in the design of active queue management mechanisms," in *Proceedings of the 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 610–615, Opatija, Croatia, 2015.
- [21] J. F. Kingman, *Poisson processes*, Oxford University Press, 1993.
- [22] J. W. Cohen, *The single server queue*, North-Holland Publishing Company, Amsterdam, Second edition, 1982.
- [23] M. Muresan, *A Concrete Approach to Classical Analysis*, Springer, Berlin, Germany, 2009.
- [24] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," in *Proceedings of the ITC Conference on IP Traffic, Modeling and Management*, pp. 282–289, 2000.
- [25] P. Benko and A. Veres, "A passive method for estimating end-to-end TCP packet loss," in *Proceedings of the IEEE GLOBECOM'02*, pp. 2609–2613, Taiwan, 2002.
- [26] N. G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *Proceedings IEEE INFOCOM 2001*, pp. 915–923, Canada, USA, 2001.
- [27] J. Sommers, P. Barford, N. Duffield, and A. Ron, "Improving accuracy in end-to-end packet loss measurement," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 157–168, 2005.
- [28] A. Chydzinski, R. Wojcicki, and G. Hryn, "On the number of losses in an MMPP queue," *Lecture Notes in Computer Science*, vol. 4712, pp. 38–48, 2007.
- [29] H. Takagi, *Queueing analysis - Finite Systems*, North-Holland, Amsterdam, 1993.
- [30] J. W. McGowan, "Burst ratio: a measure of bursty loss on packet-based networks," *US Patent*, 2005.
- [31] A. Chydzinski and D. Samociuk, "Burst ratio in a single-server queue," *Telecommunication Systems*, pp. 1–14, 2018.
- [32] A. Chydzinski, D. Samociuk, and B. Adamczyk, "Burst ratio in the finite-buffer queue with batch Poisson arrivals," *Applied Mathematics and Computation*, vol. 330, pp. 225–238, 2018.

- [33] M. Zukerman and I. Rubin, "Performance of flow-controlled communications systems under bursty traffic," in *Proceedings of IEEE GLOBECOM86*, vol. 3, pp. 1266–1271, Houston, Texas, USA, 1986.
- [34] ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning," Tech. Rep., 2014.
- [35] D. Samociuk and A. Chydzinski, "On the impact of the dropping function on the packet queueing performance," in *Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 473–478, IEEE, Opatija, Croatia, 2018.
- [36] <https://omnetpp.org>.



Hindawi

Submit your manuscripts at
www.hindawi.com

