Impact of the Dropping Function on Clustering of Packet Losses

Andrzej Chydzinski

Abstract: The dropping function mechanism is known to improve the performance of TCP/IP networks by reducing queueing delays and desynchronizing flows. In this paper, we study yet another positive effect caused by this mechanism, i.e., the reduction in the clustering of packet losses, measured by the burst ratio. The main contribution consists of two new formulas for the burst ratio in systems with and without the dropping function, respectively. These formulas enable the easy calculation of the burst ratio for a general, non-Poisson traffic, and for an arbitrary form of the dropping function. Having the formulas, we provide several numerical examples that demonstrate their usability. In particular, we test the effect of the dropping function’s shape on the burst ratio. Several shapes of the dropping function proposed in the literature are compared in this context. We also demonstrate, how the optimal shape can be found in a parameter-depended class of functions. Finally, we investigate the impact of different system parameters on the burst ratio, including the load of the system and the variance of the service time. The most important conclusion drawn from these examples is that it is not only the dropping function that reduces the burst ratio by far; simultaneously, the more variable the traffic, the more beneficial the application of the dropping function.

Keywords: TCP/IP networks; bufferbloat; dropping function; packet losses; burst ratio; queueing model

1. Introduction

The basic queueing mechanism used commonly for packet queuing in Internet routers, i.e., the tail-drop FIFO queue, has several important drawbacks. First of all, it causes the infamous bufferbloat effect, meaning that there are long queues of packets in a significant fraction of buffers in the Internet. This not only adds substantial delays to packet delivery times but has also other negative consequences, such as the synchronization of TCP control loops among flows, unfair bandwidth allocation for flows with long RTT times, and others [1].

Recently, yet another disadvantage of simple tail-drop queueing was discovered and described. Namely, when the router’s buffer is full, several incoming packets can be dropped in a row, forming a sequence of losses [2,3]. Such sequences of losses are particularly bad for multimedia streams of real-time character, e.g., voice or video calls.

The solution of the bufferbloat problem, advocated strongly by the Internet Engineering Task Force, is the application of active queue management (AQM) in Internet routers [1]. In AQM, contrary to the tail-drop mechanism, an incoming packet can be dropped even if the router’s buffer is not full. Actually, it can be dropped even if the buffer occupancy is rather low. Moreover, the decision of whether an incoming packet should be accepted or dropped is probabilistic in nature, i.e., each packet can be accepted with some, dynamically changing, probability. Such randomization is supposed to mitigate the synchronization of TCP flows.

There are many active queue management algorithms proposed to date (see, e.g., [4–8] and the references given there). The main difference between them is how the dropping probability is computed based on current and past system parameters, such as the queue length and its dynamics, the packet loss events, the empty-system events, and others.
In some AQM algorithms, advanced methods are exploited to compute the dropping probability, including neural networks [9,10], genetic algorithms [11], and fuzzy logic [12].

An important subclass of AQM algorithms is based on the dropping function. In such algorithms, the packet dropping probability is a function of the length of the queue of packets in the buffer. In fact, historically, the first AQM algorithm, RED [13], belongs to this class. In RED, a simple linear dropping function was used. After that, several other shapes of the dropping function have been tested, including the doubly linear shape [14], negative-exponential shape [15], quadratic shape [16], a mixture of third-degree polynomials and linear functions [17], and a product of logarithmic and linear functions [18].

The dropping function mechanism may not provide such excellent results as the mentioned methods based on machine learning techniques. It is, however, still worth our attention for at least two reasons. Firstly, it is very simple to implement in a networking device, contrary to advanced algorithms. Secondly, it still provides great improvements, if compared to the tail-drop mechanism.

There are no commercially available networking devices with the built-in dropping function mechanism. However, the mechanism has been recently implemented in an experimental device and tested in a real network [19]. Namely, the dropping function has been implemented in an ×86-multi-core server with DPDK cards and software, which enabled packet processing and forwarding speeds exceeding those required by the network itself. Then, the device was tested for over a month in the network of a large university, operating as usual. During this period, several thousands of performance measurements were gathered, at different times of the day and night and at different days within the week and weekend, i.e., under different traffic intensities and types. As shown in [19], various shapes of the dropping function provide significant shortening of queues and improvements in their stability while maintaining the packet loss rate at a similar level, as in the tail-drop case. Yet another observation made in [19] is a significant reduction in the burst ratio parameter by the dropping function. The burst ratio is a parameter that expresses how many times an average sequence of losses in the observed stream of packets is longer than an average sequence of losses in a hypothetical stream, in which all losses are uncorrelated. Therefore, the burst ratio characterizes the strength of the tendency of losses to form long sequences, which are particularly unwelcome in real-time multimedia transmissions.

It is intuitive and easy to understand why the burst ratio can be elevated in a network exploiting tail-drop queues in its nodes. Namely, several packets can be dropped in a row during continuous periods of time when the buffer is full (buffer overflow periods). This phenomenon was confirmed by direct measurements in a lab and in a real network, [2,3]. It was also explained theoretically, by the derivation of the burst ratio for tail-drop queueing models with various arrival stream types [20–22]. It is equally easy to understand why the dropping function mechanism can decrease the burst ratio. When this mechanism is used, the buffer overflow period occurs but is extremely rare or never occurs. In other words, there are no periods of time during which all arriving packets are lost. Instead, every packet can be randomly accepted or dropped, which makes it much harder to form a sequence of consecutive losses.

Therefore, the state-of-the-art finding can be summarized as follows. It has been shown in networking experiments with a prototype device that the dropping function reduces the burst ratio significantly. Intuitively, this is expected. Thus far, however, there has been no deep mathematical analysis of this effect and no way to predict the extent of this effect precisely using numbers.

The purpose of this paper is to feel this gap, i.e., to make possible the calculation of the burst ratio in advance for a given parameterization of the traffic and a given form of the dropping function. The main contribution of this paper consists of explicite formulas for the burst ratio parameter in queueing models with the dropping function (Theorem 1) and without the dropping function (Theorem 2). What is important is that a general type of traffic, with the interarrival time distribution having an arbitrary form, is assumed. Having
proven the formulas, we first study the impact of the dropping function on the burst ratio depending on the shape of the dropping function. Several shapes of the dropping function proposed in the literature are compared. We also demonstrate how the optimal shape can be found in a parameter-dependent class of functions. Then, we investigate the influence of other system parameters on the burst ratio, including the load of the system and the standard deviation of the service time.

To the author’s best knowledge, the results of this paper are new. Previously, the analysis of the burst ratio in a system with the dropping function was carried out only under the assumption that the traffic is Poisson in nature [23]. Such an assumption simplifies the analysis but renders the model less useful. Namely, it is well-known that the traffic in networks is far from Poisson and using Poisson approximation leads to optimistic underestimations of performance characteristics. Therefore, to obtain precise results, we have to use the general distribution of the interarrival time, as stated here. Other previous studies on the burst ratio, e.g., [20–22,24,25], do not incorporate the dropping function, which is the crucial component herein.

The remainder of the article is organized as follows. In Section 2, the formal description of the queueing model is given, as well as the definition and some properties of the burst ratio. In Section 3, two main theorems on the burst ratio are proven, and they are devoted to systems with and without the dropping function, respectively. In Section 4, numerical results are presented and discussed. They are divided into a few subsections, in which impacts of the shape of the dropping function, the load of the system, and the interarrival time distribution are studied. At the end of Section 4, results of simulations performed to verify the theoretical results are shown. Finally, some concluding remarks are provided in Section 5.

2. The Model

2.1. Queueing System

We deal with the single-server queueing model with the addition of the dropping function. Namely, packets arrive to the buffer, where they are stored before service (transmission) in the arrival order. The interarrival time distribution has distribution function \( G(t) \), which can have any form. We assume only that the average interarrival time is finite.

\[
E[G] = \int_0^\infty t dG(t) < \infty.
\]  

(1)

Systems with infinite interarrival times are not interesting; they have an average queue length of zero. The buffer size (system capacity) is finite and equals \( K \), which means that the total number of packets that is present in the system cannot exceed \( K \), including the service position. When the system is full upon a packet arrival, the newly arriving packet is dropped (deleted).

Moreover, the dropping function mechanism is applied, which means that an arriving packet can be dropped upon arrival even if the buffer is not full. This happens with probability \( d(n) \), where \( n \) is the queue length upon the arrival of the new packet, including the service position. Function \( d(n) \) is called the dropping function and can assume an arbitrary form, if only \( 0 \leq d(n) \leq 1 \) for every \( 0 < n < K \), \( d(0) < 1 \) and \( d(n) = 1 \) for \( n \geq K \). Assumption \( d(0) < 1 \) excludes an uninteresting system with \( d(0) = 1 \) in which every arriving packet is deleted, and the queue is always empty. Assumption \( d(n) = 1 \) for \( n \geq K \) is equivalent to the final-buffer assumption, as stated above.

The service time is exponentially distributed with parameter \( \mu \). Standard independence assumptions are made, i.e., all interarrival times and service times are mutually independent. The queue length at time \( t \) is denoted by \( X(t) \), which includes the service
position, if occupied. We use the convention that the queue length process is left continuous, \( X(u-) = X(u) \). The load of the system is defined traditionally as follows.

\[
\rho = \frac{1}{\mu E}. \tag{2}
\]

It is easy to see that the presented model is equivalent to the \( G/M/1/K \) model in Kendall’s notation, but with the addition of the dropping function mechanism.

2.2. Clustering of Losses

The main characteristic studied in this paper is the burst ratio, which is introduced in [24].

Consider a stream of packets, some of them being deleted. The burst ratio, \( B \), is the ratio of the average length of the sequence of packets deleted in a row in the considered stream to the average length of the sequence of losses in a hypothetical stream in which all deletions happen independently, with some probability \( L \). Denoting the average length of the loss sequence in the observed stream by \( \overline{G} \) while in the hypothetical stream it is denoted by \( \overline{K} \), we have by definition the following.

\[
B = \frac{\overline{G}}{\overline{K}}. \tag{3}
\]

It is a simple matter to compute \( \overline{K} \) using the geometric series. We obtain \( \overline{K} = \frac{1}{1 - L} \).

Therefore, (3) yields the following:

\[
B = (1 - L)\overline{G}, \tag{4}
\]

where \( L \) is the overall packet loss probability.

In practice, we compute the burst ratio in two steps regardless of whether it is performed in experimental or theoretical studies. Firstly, we compute (or measure) the average length of the sequence of losses, \( \overline{G} \). Secondly, we compute (or measure) the loss probability, \( L \). Then, \( B \) is obtained from Formula (4).

Note that the two numbers, \( L \) and \( B \), provide compact yet very informative characterization of the loss process. \( L \) informs us about the overall ratio of losses, while \( B \) informs us about their tendency to cluster together in sequences. If \( B = 1 \), then there is no such tendency—the losses are random and uncorrelated. The greater \( B \) is above 1, the stronger the clustering of losses. In contemporary TCP/IP networks, \( B \) is typically significantly greater than 1 (see experimental studies of [2,3]). As it was explained in Section 1, this is caused by the tail-drop queueing mechanism.

Elevated values of both \( L \) and \( B \) have negative impact on real-time multimedia transmissions. In some cases, this effect can be measured precisely by using a strict formula. For instance, in [26], we can find a formula for the deterioration of the quality of IP voice calls as a function of \( L \) and \( B \) (see formula (7)–(29) of [26]).

3. Analysis

We will prove now two theorems on the burst ratio in systems with and without the dropping function, respectively.

\[ \textbf{Theorem 1.} \text{ The burst ratio in the } G/M/1/K \text{ system with dropping function } d(n) \text{ is equal to the following:} \]

\[
B = \left( 1 - \sum_{n=0}^{K} p_n d(n) \right) \frac{\sum_{j=0}^{K} r_j \Pi_j}{\sum_{j=0}^{K} r_j}, \tag{5}
\]

where

\[
r_0 = \sum_{i=0}^{K-1} p_i (1 - d(i)) c_{i+1} d(0), \tag{6}
\]
\[ r_j = \sum_{i=j-1}^{K-1} p_i (1 - d(i)) b_{i+1-j} d(j), \quad j \geq 1, \]  
\[ b_j = \frac{1}{\beta_j} \int_0^\infty e^{-\mu t} (\mu t)^j dG(t), \quad c_j = 1 - \sum_{i=0}^{j-1} b_i, \]  

The probability \( \prod_n \) is obtained recursively as follows:

\[ \prod_0 = \frac{1}{1 - d(0)}, \]  
\[ \prod_1 = \frac{1}{1 - d(1)} b_0 \left( 1 + \frac{d(0)c_1}{1 - d(0)} \right), \]  
\[ \prod_n = \frac{1}{1 - d(n)} b_0 \left( 1 + \sum_{m=1}^{n-1} d(m) \prod_n b_{n-m} + \frac{d(0)c_n}{1 - d(0)} \right), \quad n \geq 2, \]  
while

\[ (p_0, \ldots, p_K) = (1, 0, \ldots, 0) \cdot A^{-1}, \]  
where matrix \( A \) has the form:

\[ A = [a_{ij}]_{i,j=0,\ldots,K}, \quad a_{ij} = \begin{cases} 
1, & \text{if } 0 \leq i \leq K, j = 0, \\
 b_{j-i+1} (1 - d(i)) + b_{j-i} d(i), & \text{if } 0 < i \leq K, 0 < j < i, \\
b_1 (1 - d(i)) + b_0 d(i) - 1, & \text{if } 0 < i \leq K, j = i, \\
b_0 (1 - d(i)), & \text{if } 0 \leq i \leq K, j = i + 1, \\
0, & \text{otherwise.} 
\end{cases} \]  

**Proof.** Proof of Theorem 1. Let \( t_1, t_2, \ldots \) denote consecutive arrival times and \( X_l = X(t_l) \), \( l = 1, 2, \ldots \). Sequence \( \{X_l\} \) constitutes a discrete-time Markov chain. This follows from the memorylessness property of the exponential distribution; namely, the remaining service time and the packet arrival are exponentially distributed with parameter \( \mu \) no matter how much time the current service has already taken.

Take an arbitrary \( k > 1 \) and assume that the queue length is \( n \) at time \( t_k \). Let \( \prod_n \) denote the average number of consecutive losses from time \( t_k \) under the condition that the packet arriving at time \( t_k \) is lost. It must hold \( \prod_n \geq 1 \), because the packet lost at time \( t_k \) is included in \( \prod_n \). Moreover, due to the fact that \( \{X_l\} \) is a Markov chain, the evolution of the system from time \( t_k \) depends only on the queue length at time \( t_k \) and does not depend on previous queue’s lengths. Therefore, \( \prod_n \) depends only on \( n \) and does not depend on \( k \).

In the first part of the proof, we will prove Formulas (9)–(11) for \( \prod_n \). If \( n > 0 \), then we have the following.

\[ \prod_n = 1 + \sum_{m=1}^{n} d(m) \prod_n \int_0^\infty e^{-\mu t} (\mu t)^{n-m} \frac{dG(t)}{(n-m)!}, \]  
\[ + d(0) \prod_0 \sum_{i=n}^{\infty} e^{-\mu t} \frac{t^i}{i!} dG(t), \quad 0 < n \leq K. \]  

Equation (14) can be explained as follows. Number 1 stands for the packet loss at time \( t_k \). Let \( m \) be the queue length upon the next arrival time, \( t_{k+1} \). If \( m \geq 1 \), then the probability of having the queue length \( m \) at time \( t_{k+1} \) is given by the first integral in (14). This integral is obtained by conditioning on the duration of the interarrival time and using the Poisson formula for the probability of \( n - m \) completed services by time \( t_{k+1} \). Then, the packet arriving at time \( t_{k+1} \) can be lost with probability \( d(m) \) and the average number of consecutive losses from time \( t_{k+1} \) is then \( \prod_m \). This explains the first row of (14). The second row corresponds to the case \( m = 0 \). The probability of having queue length 0 at time \( t_{k+1} \) is
expressed now by the sum of integrals. Then, the packet arriving at time \( t_{k+1} \) can be lost with probability \( d(0) \) and the average number of consecutive losses from time \( t_{k+1} \) is \( \Pi_0 \).

If \( n = 0 \), then we obtain the following:

\[
\Pi_0 = 1 + d(0) \Pi_0.
\]

(15)

Again, 1 stands for the packet loss at time \( t_k \). The queue length just before the next arrival time must be 0; thus, the packet arriving at time \( t_{k+1} \) is lost with probability \( d(0) \) and the average number of consecutive losses from time \( t_{k+1} \) is \( \Pi_0 \) again.

From (15), we immediately obtain (9). Exploiting (9) and notation (8), Equation (14) can be rewritten as follows:

\[
\Pi_n = 1 + \sum_{m=1}^{n} d(m) \Pi_m b_{n-m} + \frac{d(0) c_n}{1 - d(0)}, \quad 0 < n \leq K,
\]

(16)

which for \( n = 1 \), gives the following:

\[
\Pi_1 = 1 + d(1) \Pi_0 b_0 + \frac{d(0) c_1}{1 - d(0)}.
\]

(17)

while for \( n \geq 2 \), the following is obtained.

\[
\Pi_n = 1 + d(n) \Pi_n b_0 + \sum_{m=1}^{n-1} d(m) \Pi_m b_{n-m} + \frac{d(0) c_n}{1 - d(0)}.
\]

(18)

Finally, from (17) and (18), we obtain (10) and (11), respectively.

In the second part of the proof, we will derive the overall loss probability, \( L \).

We start with computing the transition matrix for the Markov chain \( \{X_k\} \). Firstly, the probability of the transition from non-zero state \( i \) to non-zero state \( j \leq i \) is equal to the following: \( (1 - d(i)) b_{i-j+1} + d(i) b_{i-j} \).

Indeed, such a transition can happen in two ways—either the first packet is accepted and \( i - j + 1 \) services are completed by the next arrival time, or the first packet is dropped and \( i - j \) services are completed by the next arrival time. Secondly, the probability of the transition from state \( i < K \) to state \( i + 1 \) equals the following: \( (1 - d(i)) b_0 \).

Indeed, the first packet must be accepted and no service can be completed by the next arrival time in this case. Finally, the probability of the transition from any state to state 0 is as follows.

\[
(1 - d(i))(1 - \sum_{j=0}^{i} b_j) + d(i)(1 - \sum_{j=0}^{i-1} b_j).
\]

Other transitions are not possible. Summarizing these considerations, we obtain the transition matrix \( Q \) in the following form:

\[
Q = [q_{ij}]_{i,j=0,...,K}, \quad q_{ij} = \begin{cases} c_{i+1}(1 - d(i)) + c_i d(i), & \text{if } 0 \leq i \leq K, j = 0, \\ b_{j-i+1}(1 - d(i)) + b_j d(i), & \text{if } 0 < i \leq K, 0 < j \leq i, \\ b_0 (1 - d(i)), & \text{if } 0 \leq i \leq K - 1, j = i + 1, \\ 0, & \text{otherwise.} \end{cases}
\]

(19)

The stationary vector \( p = (p_0, \ldots, p_K) \) for this chain can be obtained in the standard method by solving the system of equations:

\[
\begin{cases} p Q = p, \\ \sum_{j=0}^{K} p_j = 1. \end{cases}
\]

(20)
Matrix $Q$ is known to be linearly dependent; thus, the Equation in (20) corresponding to the first column of $Q$ can be removed. Then, rearranging the remaining equations and grouping $(p_0, \ldots, p_K)$ on the left side, we obtain an explicit solution of (20) in (12) and (13).

Now, due to the fact that $p_n$ is the stationary probability that the queue length upon a packet arrival is $n$, we can compute the loss probability as follows.

$$L = \sum_{n=0}^{K} p_n d(n).$$  \hfill (21)

In the third part of the proof, we will derive the average length of the sequence of consecutive losses, $\mathcal{C}$, using $\mathcal{P}_n$. Note first that it is not quite trivial to obtain $\mathcal{C}$ from $\mathcal{P}_n$, because in the definition of $\mathcal{P}_n$, it is not assumed that the sequence of losses begins at time $t_k$—it may begin before $t_k$.

To overcome this, consider a sequence of losses that begins at arrival time $t_k$, when the system is in the stationary regime and $X_k = j$. Such a sequence can be initiated only if the previous packet was accepted to the buffer. In particular, the previous packet must have arrived to the buffer at time $t_{k-1}$, when the queue length was $i$ and $j - 1 \leq i < K$, it must have been accepted with probability $1 - d(i)$, and the queue length must have changed from $i$ to $j$ during the interarrival time. It is easy to see that for $j > 0$, such a series of events happens with the following probability:

$$p_i (1 - d(i)) b_{i+1-j} d(j),$$ \hfill (22)

while for $j = 0$, such a series of events happens with the following probability.

$$p_i (1 - d(i)) c_{i+1} d(0).$$ \hfill (23)

Now, define $r_j$ as the probability that, at an arbitrary arrival time, the queue length is $j$ and a sequence of losses begins at that time. From the considerations of the previous paragraph and (22) and (23), we obtain Formulas (7) and (6) for $r_j$, respectively. The probability that a sequence of losses begins at arbitrary arrival time is then $\sum_{j=0}^{K} r_j$. Therefore, we can conclude that the average length of a sequence of losses, which begins at arbitrary arrival time, is as follows.

$$\mathcal{C} = \frac{\sum_{j=0}^{K} r_j \mathcal{P}_j}{\sum_{j=0}^{K} r_j}. \hfill (24)$$

Finally, combining (4) with (21) and (24), we arrive at (5), which completes the proof. \qed

Now we can prove an analog of Theorem 1 but for the system without the dropping function.

**Theorem 2.** The burst ratio in the G/M/1/K system is equal to the following:

$$B = \frac{1 - P_K}{1 - P_0},$$ \hfill (25)

where $b_j$ is given in (8) while the following is the case:

$$(p_0, \ldots, p_K) = (1, 0, \ldots, 0) \cdot A^{-1},$$ \hfill (26)
and matrix $A$ is provided as follows:

$$A = [a_{ij}]_{i,j=0,...,K}, \quad a_{ij} = \begin{cases} 
1, & \text{if } 0 \leq i \leq K, j = 0, \\
\frac{b_{i-j+1}}{K}, & \text{if } 0 < i < K, 0 < j < i, \\
\frac{b_{K-j}}{K}, & \text{if } i = K, 0 < j < K, \\
\frac{b_{1}}{K} - 1, & \text{if } 0 < i < K, j = i, \\
\frac{b_{0}}{K} - 1, & \text{if } i = j = K, \\
\frac{b_{0}}{K}, & \text{if } 0 \leq i \leq K - 1, j = i + 1, \\
0, & \text{otherwise}. 
\end{cases}$$ (27)

**Proof.** Proof of Theorem 2. This theorem can be proven at least in two different ways.

In one proof, we can use Theorem 1 and apply a trivial dropping function, i.e., $d(n) = 0$ for $n < K$ and $d(n) = 1$ otherwise. Obviously, such a dropping function renders the system equivalent to the system without the dropping function, but with a finite buffer. Using this trivial dropping function, we obtain the following from (6) and (7):

$$r_j = 0, \quad 0 \leq j < K,$$ (28)

while from (9) to (11), the following is obtained

$$H_n = 1, \quad 0 \leq n < K,$$ (29)

$$H_K = \frac{1}{1 - b_0},$$ (30)

and from (21), we have the following.

$$L = p_K.$$ (31)

Formulas (28)–(31) combined with (5) lead to (25), where matrix $A$ in (27) is just a simplified matrix (13).

Alternatively, the proof can be conducted without references to Theorem 1 using probabilistic arguments. We have to notice two facts. Firstly, a sequence of losses in the system can occur only during the buffer’s overflow period. Secondly, the duration of the buffer’s overflow period is exponentially distributed with parameter $\mu$. This is a consequence of the memorylessness property of the exponential distribution, i.e., no matter when the buffer is overflowed, the remaining service time is exponentially distributed. On the other hand, the buffer overflow period is equal to the remaining service time upon the buffer’s overflow. Now, consider the beginning of the overflow period, i.e., the arrival time, when the queue length jumps from $K - 1$ to $K$. With probability $b_0$, the next arrival will happen before the end of the overflow period. In this case, the new packet is lost and the sequence of losses is extended by 1. What is more, the probability of having yet another arrival before the end of the overflow period is again $b_0$ due to the memorylessness property of the exponential distribution. Therefore, we have in fact a series of Bernoulli experiments, in which the probability of a failure (loss) in a single experiment is $b_0$. Therefore, the average length of a sequence of failures (losses) is equal to $\frac{1}{1 - b_0}$. This, combined with the obvious relation $L = p_K$ and (4), gives (25), while matrix $A$ can be easily obtained by using transition probabilities of chain $\{X_t\}$. \(\square\)

4. Examples

In the numerical examples, we use the following forms of the dropping function.

$$d_1(n) = \begin{cases} 
0, & \text{if } n < 20, \\
0.005n - 0.1, & \text{if } 20 \leq n < 40, \\
0.09n - 3.5, & \text{if } 40 \leq n < 50, \\
1, & \text{if } n \geq 50, 
\end{cases}$$
They are also depicted in Figure 1. As observed, the doubly linear, quadratic, linear, combined cubic, and negative exponential shapes of the dropping function are used. Such shapes of the dropping function are not new. They were proposed in the literature by the authors of [13–17]. In none of these papers, however, was there a study of the impact of the dropping function on the burst ratio carried out. The particular parameterizations are based on the assumption that the dropping function operates roughly at the second half of the buffer size. Assuming a buffer size of 50 packets, the dropping function operates herein on interval [20,50]. In every case, the dropping begins with zero probability at point (20,0), and ends with the probability of 1 at point (50,1). This enables obtaining sample parameterizations of the dropping function as presented above.

![Figure 1. Dropping functions $d_1$–$d_5$.](image)

As it was already mentioned, in every case with the dropping function, the buffer size is $K = 50$. The same buffer size is used in examples with no dropping function. If not stated otherwise, the interarrival time distribution is gamma with parameters $\alpha = \beta = 0.25$, i.e., with the mean of 1 and the standard deviation of 2. This is altered in Section 4.3, where the impact of the interarrival time on the burst ratio is studied and in simulations shown in Section 4.4. Similarly, if not stated otherwise, the service rate is $\mu = 1$, resulting in $\rho = 1$. This is altered in Section 4.2, where the influence of the system load is investigated and in simulations in Section 4.4.
4.1. Impact of \(d(n)\) Shape

In this subsection, we will check the influence of the shape and aggressiveness of the dropping function on the burst ratio.

In Table 1, burst ratios \(B_1 - B_5\) computed for functions \(d_1(n) - d_5(n)\) are presented, as well as the burst ratio computed without the dropping function, \(B_0\). Additionally, every \(B_i\) is presented as a percentage of \(B_0\). The numbers were obtained using Theorems 1 and 2.

**Table 1.** Burst ratio \(B_i\) for \(i\)-th dropping function in the first row. In the second row, \(B_i\) is a percentage of \(B_0\).

| Dropping Function | None | \(d_1(n)\) | \(d_2(n)\) | \(d_3(n)\) | \(d_4(n)\) | \(d_5(n)\) |
|-------------------|------|------------|------------|------------|------------|------------|
| \(B_i\)          | 2.8764 | 1.3236     | 1.2746     | 1.2839     | 1.3634     | 1.4389     |
| \(B_i / B_0\)    | 100%  | 46%        | 44%        | 45%        | 47%        | 50%        |

The following conclusions can be drawn from Table 1. Firstly, a high burst ratio is observed in the case with no dropping function. Secondly, the application of the dropping function reduces its value significantly. Namely, \(B_0\) is reduced to 44–50% of its original value, when the dropping function is used. There are some differences among \(d_1 - d_5\), but all of them provide a great improvement.

To analyze in detail the impact of particular shape of the dropping function, we note first that \(d_1 - d_5\) are numbered roughly from the less aggressive, \(d_1\), to the most aggressive, \(d_5\) (see Figure 1). The best burst ratio is obtained for a rather mild function \(d_2\). The general picture is, however, rather complicated. Function \(d_1\) performs worse than \(d_2\), even though \(d_1\) is less aggressive in general. One can argue that this is because \(d_1\) is more aggressive in the initial interval \([20,23]\). On the other hand, \(d_4\) is even less aggressive than \(d_2\) in this interval, but it provides worse \(B\). It is also interesting to compare the results computed for \(d_3\) and \(d_4\), because the latter is less aggressive in the interval \([20,30]\) than the former, the same in the interval \([30,40]\), and more aggressive in the interval \([40,50]\). As we see in Table 1, \(d_3\) performs better than \(d_4\).

To study the impact of the dropping function shape more systematically, we can use a class of functions \(d_v^3(n)\), where exponent \(v\) is a parameter. A few selected shapes of function \(d_v^3\), for \(v\) from 0.1 to 20, are presented in Figure 2.

**Figure 2.** Dropping functions \(d_v^3(n)\) for \(v\) from 0.1 to 20.

In Figure 3 and Table 2, the burst ratio as a function of parameter \(v\) is shown.
Table 2. Burst ratio for selected values of parameter $v$.

| $v$   | 0      | 0.1    | 0.5    | 1.5    | 5      | 10     | 50     | 100    | $\infty$ |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|----------|
| $B$   | 2.6906 | 2.015  | 1.3953 | 1.2674 | 1.4047 | 1.6360 | 2.5622 | 2.8186 | 2.8764   |

Figure 3. Burst ratio versus parameter $v$ when dropping function $d_3^v(n)$ is applied.

It is easy to observe that if $v \to \infty$, then function $d_3^v(n)$ converges to the unit-step function with the step at $n = 50$. Therefore, we can see in Figure 3 that the burst ratio approaches a limit when $v \to \infty$. The limiting value is 2.8764 (see Table 2), and it is equal to the burst ratio in a system without the dropping function and the buffer of size 50. Similarly, if $v \to 0^+$, then function $d_3^v(n)$ converges to the unit-step function with the step at $n = 20$. Therefore, if $v \to 0^+$, then the burst approaches the value for a system without the dropping function and the buffer of size 20. This value is 2.6906; see Table 2.

What is important is that we can observe a minimum for $v = 1.5$ in Figure 3. The optimal value of the burst ratio, achieved for $v = 1.5$, is equal to 1.2674 (see Table 2). The shape of the dropping function that provides the minimal value of the burst ratio is depicted in Figure 2 in red.

Now, it is hard to explain why this particular dropping function in class $d_3^v$ provides the best burst ratio. In Theorem 1, we see that the burst ratio depends in a very complicated way on the dropping function. The form of this function influences the stationary distribution of the queue length, the loss probability and the structure of losses, each of them having impact on each other and on the resulting burst ratio. Fortunately, we can always use Theorem 1 to find the optimal shape of the dropping function with respect to the burst ratio in a parameter-dependent class of functions, as shown above.

4.2. Impact of $\rho$

In the next set of numerical results, we study the influence of the system load on the burst ratio. By varying the service rate, $\mu$, we change the load of the system from a severely underloaded one ($\rho = 0.5$) to a severely overloaded one ($\rho = 2.0$) and check how it influences the burst ratio when dropping functions $d_1$–$d_5$ are applied, as well as when there is no dropping function.

The results are presented in Figure 4 and Table 3. As we can notice, a high value of the burst ratio is achieved in a system with no dropping function, in the entire interval of the load, from 0.5 to 2.0. Moreover, there is a maximum on the black curve in Figure 4 close to $\rho = 1$. When any form of the dropping function is applied, the burst ratio is reduced significantly, again in the entire interval, which is clearly seen in both Figure 4 and Table 3.
Table 3. Burst ratio for selected values of the system’s load.

| $\rho$   | $\rho = 0.5$ | $\rho = 0.75$ | $\rho = 1$ | $\rho = 1.25$ | $\rho = 1.5$ | $\rho = 1.75$ | $\rho = 2$ |
|---------|--------------|--------------|------------|--------------|--------------|--------------|------------|
| no drop. fun. | 2.3660 | 2.7029 | 2.8764 | 2.6484 | 2.4036 | 2.2212 | 2.0819 |
| $d_1(n)$ | 1.0383 | 1.1499 | 1.2326 | 1.3390 | 1.2904 | 1.2442 | 1.1687 |
| $d_2(n)$ | 1.0947 | 1.1989 | 1.2746 | 1.3261 | 1.1794 | 1.1412 | 1.1160 |
| $d_3(n)$ | 1.1566 | 1.2455 | 1.2839 | 1.2287 | 1.1616 | 1.1167 | 1.0888 |
| $d_4(n)$ | 1.2006 | 1.3174 | 1.3634 | 1.2903 | 1.2049 | 1.1477 | 1.1122 |
| $d_5(n)$ | 1.2999 | 1.4104 | 1.4389 | 1.3633 | 1.2700 | 1.1987 | 1.1488 |

Figure 4. Burst ratio versus the system load for dropping functions $d_1$–$d_5$ and no dropping function.

In Table 4, $B_i$ as a percentage of $B_0$ is presented. As we can see, the application of the dropping reduces the burst ratio to 43–58% of its original value, depending on the dropping function form and the system load.

Table 4. Burst ratio $B_i$ as a percentage of $B_0$ for different values of the system load.

| $\rho$   | $\rho = 0.5$ | $\rho = 0.75$ | $\rho = 1$ | $\rho = 1.25$ | $\rho = 1.5$ | $\rho = 1.75$ | $\rho = 2$ |
|---------|--------------|--------------|------------|--------------|--------------|--------------|------------|
| $B_1/B_0$ | 44% | 43% | 46% | 51% | 54% | 56% | 58% |
| $B_2/B_0$ | 46% | 44% | 44% | 47% | 49% | 51% | 54% |
| $B_3/B_0$ | 49% | 46% | 45% | 46% | 48% | 50% | 52% |
| $B_4/B_0$ | 51% | 49% | 47% | 49% | 50% | 52% | 53% |
| $B_5/B_0$ | 55% | 52% | 50% | 51% | 53% | 54% | 55% |

Two more interesting observations can be made.

Firstly, the curves for $d_1$–$d_5$ in Figure 4 intersect a few times. This means that it depends on the system load, and when shaped among $d_1$–$d_5$, it performs better; otherwise, it performs worse. For $\rho = 0.5$, in particular, $d_1$ is the best, followed by $d_2$, $d_3$, $d_4$, and $d_5$. For $\rho = 1$, $d_2$ is the best, followed by $d_3$, $d_1$, $d_4$, and $d_5$. For $\rho = 2$, $d_3$ is the best, followed by $d_4$, $d_2$, $d_5$, and $d_1$. Therefore, we cannot say that one form of the dropping function is better than the other in general—their performance may vary with $\rho$, which may change in TCP/IP networks. Fortunately, all functions $d_1$–$d_5$ provide similarly low values of $B$ when compared to the system with no dropping function in the entire range of $\rho$. Hence, which particular shape is applied is not that important. It is important to apply one of them.

Second, if the dropping function is applied, then the local maximum of the burst ratio can be assumed for $\rho$ and not necessarily close to 1. For instance, we see in Figure 4 that in the case of $d_5$, the maximum is achieved for $\rho < 1$, while in the case of $d_1$, for $\rho > 1$. Similarly, in Table 3 we see that the burst ratio for $\rho = 1.25$ is greater than for $\rho = 1$.

Finding analytically the maximum of the burst ratio as a function of $\rho$ seems to be very hard. Fortunately, this maximum does not have a great practical meaning, because all
curves for dropping functions $d_1$–$d_5$ are rather flat. Moreover, in practice, we do not have such large loads, such as 2. Thus, the interval of interest is shorter, making the curves even flatter. Finally, there is no need to analyze the burst ratio for a very small load, e.g., 0.1. In such cases, losses are very rare, so the value of the burst ratio is practically insignificant.

4.3. Impact of $DG$

In this set of numerical results, we study the impact of the interarrival time distribution on the burst ratio. We focus on the value of the standard deviation of the interarrival time, $DG$ (varying the average interarrival time, $EG$, changes the system load, for which its influence was already studied).

Therefore, we use the gamma distribution of the interarrival time, with $\alpha = \beta = w$, where $w > 0$ is a parameter. Hence, $EG = 1$ for every $w$, while $DG$ is a function of $w$, namely $DG = w^{-1/2}$.

The results are presented in Figure 5 and Table 5. They were obtained by varying $w$ in such a way that the resulting $DG$ varied from 0 to 10.

![Figure 5. Burst ratio versus the standard deviation of the interarrival time for dropping functions $d_1$–$d_5$ and no dropping function.](image)

| $DG$ | $D_0$ | $D_2$ | $D_4$ | $D_6$ | $D_8$ | $D_{10}$ |
|------|-------|-------|-------|-------|-------|---------|
| no drop. fun. | 1.5662 | 2.8764 | 5.3032 | 7.8608 | 10.2240 | 12.3063 |
| $d_1(n)$ | 1.0629 | 1.3236 | 1.8112 | 2.2706 | 2.6652 | 2.9969 |
| $d_2(n)$ | 1.0987 | 1.2746 | 1.5621 | 1.8400 | 2.0836 | 2.2907 |
| $d_3(n)$ | 1.1313 | 1.2839 | 1.4761 | 1.6377 | 1.7716 | 1.8832 |
| $d_4(n)$ | 1.1569 | 1.3634 | 1.6010 | 1.8218 | 2.0098 | 2.1609 |
| $d_5(n)$ | 1.2071 | 1.4389 | 1.6685 | 1.8043 | 1.8837 | 1.9317 |

As we can see in Figure 5, the burst ratio depends strongly on the standard deviation of the interarrival time. In the case with no dropping function and highly variable traffic, large values of the burst ratio are obtained above 12.3 (see Table 5). This means that the average sequence of losses is 12.3-times longer than in the case of uncorrelated losses.

Fortunately, every considered dropping function decreases the burst ratio significantly. Moreover, the larger the $DG$, the greater the relative improvement of the burst ratio. This is demonstrated in Figure 6, in which $B_i$ as a percentage of $B_0$ is depicted. For instance, if $DG = 2$, then the burst ratio is reduced to 43–58% of its original value depending on the dropping function form. For $DG = 4$, the burst ratio is reduced to 24–28% of its original value, while for $DG = 10$, the burst ratio is reduced to 15–24% of its original value, depending on the dropping function form.
4.4. Verification via Simulations

To make sure that the theoretical results of Theorems 1 and 2 are error free, they were also verified using simulations. For this purpose, Omnet++ discrete-event simulator was exploited [27]. Tens of simulation runs were carried out, with different forms of the dropping function, interarrival time distributions and system loads. In every simulation run, 100 million simulated packets passed through the system. In every case, a high agreement between simulated and theoretical result was obtained.

In Table 6, a few of these results are presented, obtained for dropping functions $d_1$–$d_5$ and no dropping function; the interarrival time is parameterized as in Section 4.3, and the load varied from 0.9 to 1.4. As we can see, the simulated and theoretical values are very close to each other. A slightly worse result for $\rho = 0.9$ can be associated with the fact that there were relatively few losses observed in this simulation run for an obvious reason. Nonetheless, the resulting relative error is still less than $1/10^4$.

Table 6. Simulated versus theoretical burst ratios for different dropping functions, loads, and interarrival time standard deviations.

| System Parameters | Simulation | Theory |
|-------------------|------------|--------|
| no drop, $f_i$, $\rho = 0.9$, $DG = 4.0$ | 5.3724 | 5.3719 |
| $d_1(n)$, $\rho = 1.0$, $DG = 3.0$ | 1.5649 | 1.5647 |
| $d_2(n)$, $\rho = 1.1$, $DG = 2.0$ | 1.2689 | 1.2688 |
| $d_3(n)$, $\rho = 1.2$, $DG = 1.0$ | 1.1316 | 1.1316 |
| $d_4(n)$, $\rho = 1.3$, $DG = 0.5$ | 1.0972 | 1.0972 |
| $d_5(n)$, $\rho = 1.4$, $DG = 0.25$ | 1.0924 | 1.0924 |

5. Conclusions

In this paper, an analysis of the burst ratio parameter in a queueing system with the dropping function was carried out.

The study was motivated by two facts. Firstly, the dropping function is one of the important tools of choice in active queue management, which is recommended by IETF to reduce queueing delays and to desynchronize TCP control loops in the Internet. Secondly, it has been observed in recent experiments that the application of the dropping function reduces the burst ratio as well, which is a nice side effect. The burst ratio reflects the tendency of packet losses to form long sequences, which are especially bad for real-time multimedia transmissions.

The main contribution of this paper consists of two new theorems on the value of the burst ratio. They enable an easy calculation of the burst ratio value in systems with an arbitrary distribution of the interarrival time, an arbitrary form of the dropping function,
and with no dropping function. They enable also finding the optimal shape of the dropping function, in terms of the minimal burst ratio, in a parameter-dependent class of functions.

Theoretical results were illustrated with numerical examples based on several dropping function types proposed in the literature. In particular, the influence of the dropping function shape, the system load, and the standard deviation of the service time on the burst ratio was investigated.

At least three important observations were made.

First, the application of the dropping function reduced the burst ratio significantly in all considered scenarios. In most scenarios, with moderate variability of the interarrival time, the reduced burst burst ratio was about 50% smaller than the original burst ratio obtained without the dropping function (see Tables 1 and 4).

Second, the worse was the traffic in terms of the variability of the interarrival time: the more variable the interarrival time, the more beneficial the application of the dropping function, i.e., the greater the reduction. For instance, when the coefficient of variation was 10, the burst ratio reduced to 15–24% of its original value (see Figure 6).

Third, the differences in the performance offered by different shapes of the dropping function, proposed in the literature, were not significant—all of them provided a similar, high reduction of the burst ratio when compared to no dropping function.

**Funding:** This research was funded by National Science Centre, Poland, grant number 2020/39/B/ST6/00224.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The author declares no conflict of interest.

**References**

1. Internet Engineering Task Force. *Request for Comments 7567*; Baker, F., Fairhurst, G., Eds.; 2015; ISSN 2070-1721. Available online: https://datatracker.ietf.org/doc/pdf/rfc7567 (accessed on 18 August 2022).

2. Samociuk, D.; Chydzinski, A.; Barczyk, M. Experimental Measurements of the Packet Burst Ratio Parameter. In *Communications in Computer and Information Science, Poznan, Poland, 18–20 September 2018*; Springer: Cham, Switzerland, 2018; Volume 928, pp. 455–466.

3. Samociuk, D.; Chydzinski, A.; Barczyk, M. Measuring and Analyzing the Burst Ratio in IP Traffic. *Broadband Commun. Netw. Syst. 2019*, 303, 86–101.

4. Khoshnevisan, L.; Salmasi, F.R. A robust and high-performance queue management controller for large round trip time networks. *Int. J. Syst. Sci. 2016*, 47, 1–12. [CrossRef]

5. Wang, P.; Zhu, D.; Lu, X. Active queue management algorithm based on data-driven predictive control. *Telecommun. Syst. 2017*, 64, 1–9. [CrossRef]

6. Kahe, G.; Jahangir, A.H. A self-tuning controller for queuing delay regulation in TCP/AQM networks. *Telecommun. Syst. 2019*, 71, 215–229. [CrossRef]

7. Bisoy, S.K.; Pattnaik, P.K.; Sain, M.; Jeong, D.U. A Self-Tuning Congestion Tracking Control for TCP/AQM Network for Single and Multiple Bottleneck Topology. *IEEE Access 2021*, 9, 27723–27735. [CrossRef]

8. Menacer, O.; Messai, A.; Kassa-Baghdouche, L. Improved Variable Structure Proportional-Integral Controller for TCP /AQM Network Systems. *J. Electr. Eng. Technol. 2021*, 16, 2235–2243. [CrossRef]

9. Chrost, L.; Chydzinski, A. On the deterministic approach to active queue management. *Telecommun. Syst. 2016*, 63, 27–44. [CrossRef]

10. Wang, C.; Chen, X.; Cao, J.; Qiu, J.; Liu, Y.; Luo, Y. Neural Network-Based Distributed Adaptive Pre-Assigned Finite-Time Consensus of Multiple TCP/AQM Networks. *IEEE Trans. Circuits Syst. 2021*, 68, 387–395. [CrossRef]

11. Al-Faiz, M.Z. Optimal linear quadratic controller based on genetic algorithm for TCP/AQM router. In Proceedings of the International Conference on Future Communication Networks (ICFCN), Niagara Falls, ON, Canada, 9–11 August 2022; pp. 78–83.

12. Chen, J.V.; Chen, F.C.; Tarn, J.M.; Yen, D.C. Improving network congestion: A RED-based fuzzy PID approach. *Comput. Stand. Interfaces 2012*, 34, 426–438. [CrossRef]

13. Floyd, S.; Jacobson, V. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw. 1993*, 1, 397–413. [CrossRef]

14. Rosolen, V.; Bonaventure, O.; Leduc, G. A RED discard strategy for ATM networks and its performance evaluation with TCP/IP traffic. *ACM Sigcomm Comput. Commun. Rev. 1999*, 29, 23–43. [CrossRef]
15. Athuraliya, S.; Li, V.H.; Low, S.H.; Yin, Q. REM: Active Queue Management. *IEEE Netw.* 2001, 15, 48–53. [CrossRef]
16. Zhou, K.; Yeung, K.L.; Li, V. Nonlinear RED: A simple yet efficient active queue management scheme. *Comput. Netw.* 2006, 50, 3784–3794. [CrossRef]
17. Feng, C.; Huang, L.; Xu, C.; Chang, Y. Congestion Control Scheme Performance Analysis Based on Nonlinear RED. *IEEE Syst. J.* 2017, 11, 2247–2254. [CrossRef]
18. Patel, S.; Karmeshu. A New Modified Dropping Function for Congested AQM Networks. *Wirel. Pers. Commun.* 2019, 104, 37–55. [CrossRef]
19. Barczyk, M.; Chydzinski, A. AQM based on the queue length: A real-network study. *PLoS ONE* 2022, 17, e0263407. [CrossRef] [PubMed]
20. Chydzinski, A.; Samociuk, D. Burst ratio in a single-server queue. *Telecommun. Syst.* 2019, 70, 263–276. [CrossRef]
21. Chydzinski, A.; Samociuk, D.; Adamczyk, B. Burst ratio in the finite-buffer queue with batch Poisson arrivals. *Appl. Math. Comput.* 2018, 330, 225–238. [CrossRef]
22. Chydzinski, A. Burst ratio for a versatile traffic model. *PLoS ONE* 2022, 17, e0272263. [CrossRef] [PubMed]
23. Chydzinski, A.; Barczyk, M.; Samociuk, D. The Single-Server Queue with the Dropping Function and Infinite Buffer. *Math. Probl. Eng.* 2018, 2018, 3260428. [CrossRef]
24. McGowan, J.W. Burst Ratio: A Measure of Bursty Loss on Packet-Based Networks. U.S. Patent 6,931,017, 2005.
25. Rachwalski, J.; Papir, Z. Analysis of Burst Ratio in Concatenated Channels. *J. Telecommun. Inf. Technol.* 2015, 65–73.
26. Bergstra, J.A.; Middelburg, C.A. ITU-T Recommendation G.107: The E-Model, a Computational Model for Use in Transmission Planning; Technical Report number G.107; International Telecommunication Union: Geneva, Switzerland, 2014.
27. OMNeT++ Discrete Event Simulator. Available online: www.omnetpp.org (accessed on 26 September 2022).