

Discrete-time buffer systems with session-based arrival streams

L. Hoflack, S. De Vuyst, S. Wittevrongel*, H. Bruneel

*SMACS Research Group, Department of Telecommunications and Information Processing,
Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium*

Abstract

The paper considers a discrete-time buffer system with infinite storage capacity and one single output channel. Users can start and end sessions during which they are active and send packets to the buffer system. In this paper we study a simple model for the resulting session-based arrival process: we assume that each active user generates a random but strictly positive number of packets per time slot. Furthermore it is assumed that the time (expressed in slots) needed to transmit a packet is geometrically distributed. The distribution of the session lengths is also geometrical. This model can be applied to study the traffic of a file server, where one file download by a user is considered to be one session. The probability generating functions of the steady-state number of active sessions, the buffer occupancy and the packet delay are derived. We also derive an approximation for the tail probabilities of the buffer occupancy. Furthermore, an expression for the mean session delay is obtained. This allows us to study the influence of the different system parameters: some examples are presented. We end by applying the model to a web server, based on actual web traffic.

Key words: discrete-time queueing model, session-based arrivals, performance evaluation, packet-based networks, web servers

1. Introduction

Packet buffers can be found in many communication systems. It is therefore important to model their behaviour to study the performance of the whole system. The performance measures of a packet buffer (buffer occupancy, delay, ...) are greatly influenced by the nature of the packet arrival process. Session-based arrival streams are a new approach for modelling the traffic streams that arise in modern telecommunication networks. We consider a user population

*Corresponding author

Email addresses: lhoflack@telin.UGent.be (L. Hoflack), sdv@telin.UGent.be (S. De Vuyst), sw@telin.UGent.be (S. Wittevrongel), hb@telin.UGent.be (H. Bruneel)

where each user is capable of starting and ending sessions. During a session a user is active and sends information through the communication system. A session can be considered as a high-level concept: e.g., a person starts surfing on the internet (start of the session) until he shuts down his computer (end of the session). Low-level interpretations are also possible: e.g., a session begins when a user starts downloading a file from a server, and the session ends when the transmission of the file is complete. In the most general session-based arrival models, the traffic generated by the users is not uniform in every time slot, but can vary (e.g. in a periodic manner, as in [20]) in time. Moreover, during one session active and non-active periods can succeed each other.

In this paper we study a more simple variant of the session-based arrival model. We consider an arrival process where each active user generates a random but strictly positive number of packets per slot. So, each user generates a non-interrupted data flow until his session ends, where the number of generated packets varies from slot to slot (but is never equal to zero). The model under study can therefore be considered as an extension of the train arrival process [7, 9, 10, 14, 15, 16, 35, 36, 37, 39], where messages (the equivalent of what we consider sessions) arrive in the buffer at the rate of one packet per slot.

Also related is the on/off arrival process (see e.g. [22, 24, 32, 40]), where a finite number of sources alternate between active periods during which one packet per slot is generated and passive periods during which no packets are generated. In [12], messages consisting of a fixed number of packets are considered in case of an uncorrelated packet arrival process. Related continuous-time models with dispersed messages are studied in [2, 11]. Note that in [2, 11, 12, 22, 24, 32, 40] the model has a finite user population (or a limitation on the maximum number of simultaneous active messages), unlike the infinite user population model assumed in the current paper where the idea of a dedicated input line per user is abandoned.

Although in our model the generated traffic does not exhibit interruptions, it has its practical use. Consider for instance a file server that serves a number of users (e.g. a video on demand server). If we define the download of a file by one user as one session, our model delivers a good description of the outgoing data buffer behaviour. As another example, consider a user hitting the ‘send’ button after writing an email. The transmission of this email to the central mail server can be considered as a session.

The purpose of this paper is to develop a basic analytical technique for the performance analysis of a discrete-time buffer system with session-based arrivals. To the best of our knowledge, such analysis technique has not been reported before. In order not to unnecessarily complicate this first analysis and to clearly present the basic approach, we have chosen to consider a geometric distribution for the session lengths in this paper. This appears to be a realistic assumption e.g. in case of small files downloaded from a web server. It is expected however that with some appropriate modifications, like a more refined state description, our analytical technique will also work for other session-length distributions.

In the next section, we give a description of the model under study. The mathematical model and the system equations are given in section 3. A detailed

analysis of the arrival process, the buffer occupancy (and its tail behaviour), the packet delay and the mean session delay is discussed in section 4. In section 5 the results are illustrated with some examples. The model is applied to an actual web server in section 6. Conclusions and future work are given in section 7.

2. Model description

We study a discrete-time queueing system with one output line and an infinite storage capacity. As usual for discrete-time models [6, 21, 33], time is divided in fixed-length time slots and transmissions can only take place at slot boundaries. Packets entering the queueing system cannot leave the buffer at the end of their arrival slot: their system time must include at least one whole time slot and their transmission can start no earlier than at the beginning of the slot following their arrival slot. So, using the terminology of [21], the system we consider in this paper is a late arrival system with delayed access (LAS-DA). An infinite user population starts and ends sessions, during which information is sent over the communication system. When a user starts a session, he generates a variable but strictly positive number of packets per slot. The session ends when the user has no more data left to send. Note that a session cannot include passive periods during which a user is idle and does not generate any information. Hence, the model is useful for the study of non-interrupted data flows, e.g. file downloads.

There is no limitation to the number of active sessions, and as we consider an infinite user population, the number of packet arrivals per slot can be very large. This means that the model can be used in situations where the bandwidth on the input side of the buffer is not an issue or where all packets originate from one source. An example of the latter is the outgoing buffer of a file server: each session is the transfer of a file to a client, but all files originate from one central place: the server's hard disk.

The numbers of new sessions starting in successive slots are assumed to be independent and identically distributed (i.i.d.) variables. In normal conditions, internet users act independently from each other and thus this seems like a realistic assumption. The session length itself is assumed to be geometrically distributed.

Finally, the transmission time of a packet is also geometrically distributed and independent of the transmission times of other packets. Note that the assumption of geometric transmission times enables us to study buffer systems for packets of variable size. Alternatively, the model can also be used to describe a system with constant packet transmission times of one slot and an unreliable output line subject to random failures that occur independently from slot to slot. Indeed, the latter results in a geometric distribution for the effective transmission times required for the successful transmission of a packet.

3. Mathematical model and system equations

We need the following definitions (k is an integer ≥ 1):

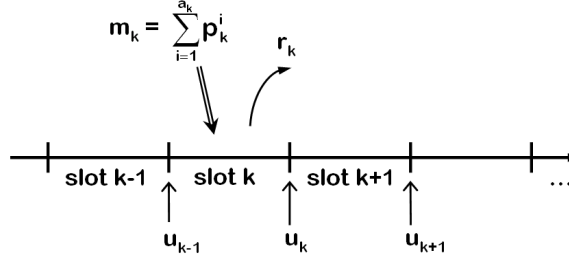


Figure 1: Definitions of the random variables m_k , u_k and r_k .

- a_k : number of active sessions during slot k ,
- s_k : number of new sessions generated during slot k ,
- u_k : buffer occupancy after slot k ,
- p_k^i : number of packets generated during slot k by session i ,
- m_k : total number of packets generated during slot k .

The probability generating function (pgf) of the number of new sessions generated per slot is given by

$$S_k(z) \triangleq E[z^{s_k}] = S(z), \quad (1)$$

where $E[\cdot]$ denotes the expected value operator.

The pgf of the number of packets generated per slot per session is given by

$$P_k^i(z) \triangleq E[z^{p_k^i}] = P(z), \quad (2)$$

where $P(0)$ equals zero, for at least one packet is generated.

The session length is geometrically distributed with parameter α , i.e. with the following probability mass function and pgf:

$$\ell(n) \triangleq \text{Prob}[\text{session length is } n \text{ slots}] = (1 - \alpha)\alpha^{n-1}, \quad n \geq 1, \quad (3)$$

$$L(z) = \frac{(1 - \alpha)z}{1 - \alpha z}. \quad (4)$$

The mean session length is therefore given by $1/(1 - \alpha)$.

The transmission time of a packet is also geometrically distributed with parameter $1 - \sigma$ and the mean transmission time equals $1/\sigma$. Owing to this geometric distribution, whenever there is a packet under transmission during a slot, this transmission will end at the end of this slot with probability σ and the transmission will continue with probability $1 - \sigma$, independent of the length of the elapsed part of the transmission time.

Let us now define a variable c_k^i that is one if the i th session that is active in slot $k-1$, still continues in slot k , and zero otherwise. As the session length is geometrically distributed, the c_k^i 's are i.i.d. Bernoulli random variables with the following probability mass function:

$$\text{Prob}[c_k^i = 0] = 1 - \alpha; \quad \text{Prob}[c_k^i = 1] = \alpha. \quad (5)$$

In view of the above model description and definitions, the evolution in time of the number of active sessions, the total number of packets generated per slot and the buffer occupancy is then expressed by the following system equations:

$$a_k = s_k + \sum_{i=1}^{a_{k-1}} c_k^i, \quad (6)$$

$$m_k = \sum_{i=1}^{a_k} p_k^i, \quad (7)$$

$$u_k = m_k + (u_{k-1} - r_k)^+, \quad (8)$$

where $(x)^+ = \max(x, 0)$ and the r_k 's are i.i.d. Bernoulli random variables with probability mass function

$$\text{Prob}[r_k = 0] = 1 - \sigma; \quad \text{Prob}[r_k = 1] = \sigma, \quad (9)$$

in view of the geometric distribution of the transmission times.

4. Steady-state queueing analysis

The state of the queueing system after slot k can be fully described by the values of two variables: the number of active sessions a_k in slot k and the buffer occupancy u_k after slot k . Due to the geometric nature of the session lengths and the transmission times, we do not need to remember the duration of the active sessions nor of the ongoing transmission: geometrically distributed variables possess the memoryless property.

We begin the analysis by defining the joint pgf of the system state after slot k (i.e. of the variables a_k and u_k) as

$$Q_k(x, z) \triangleq E[x^{a_k} z^{u_k}]. \quad (10)$$

Applying the system equations derived in the previous section, we find

$$Q_k(x, z) = S(xP(z))E\left[(C(xP(z)))^{a_{k-1}} z^{(u_{k-1} - r_k)^+}\right],$$

where $C(z) \triangleq 1 - \alpha + \alpha z$ is the pgf of the variable c_k^i . In order to further expand this equation, we need to distinguish between three cases, namely (1) $r_k = 0$, (2) $r_k = 1$, $u_{k-1} > 0$ and (3) $r_k = 1$, $u_{k-1} = 0$. Doing so, we finally get

$$Q_k(x, z) = \frac{S(xP(z))}{z} \{\Phi(z)Q_{k-1}(C(xP(z)), z) + \sigma(z-1)p_{k-1,0}\}, \quad (11)$$

where $\Phi(z) \triangleq \sigma + (1 - \sigma)z$ and $p_{k-1,0}$ denotes the probability that the buffer is empty after slot $k - 1$. Note that we used the fact that an empty buffer at the end of a slot implies that there were no active sessions during that slot, because a session generates a strictly positive number of packets per slot. Hence, $u_{k-1} = 0$ implies $a_{k-1} = 0$, which leads to $Q_{k-1}(x, 0) = Q_{k-1}(0, 0)$ for all values of x . This means that $Q_{k-1}(0, 0)$ is equal to $Q_{k-1}(1, 0) = p_{k-1,0}$.

If we let k go to infinity and assume that the system reaches a steady state, the functions Q_{k-1} and Q_k converge to the same limiting function Q . Introducing this into equation (11), we obtain that the steady-state joint pgf $Q(x, z)$ must satisfy the following functional equation:

$$Q(x, z) = \frac{S(xP(z))}{z} \{ \Phi(z)Q(C(xP(z)), z) + \sigma(z - 1)p_0 \}, \quad (12)$$

where p_0 is the steady-state probability of an empty buffer.

4.1. Arrival process

The pgf $A(x)$ of the steady-state number of active sessions can be derived by putting z equal to one in the functional equation. This leads to

$$A(x) \triangleq Q(x, 1) = S(x)A(C(x)) = \prod_{i=0}^{\infty} S(1 - \alpha^i + \alpha^i x). \quad (13)$$

An elegant proof for the convergence of (13) can be found in [31]. The mean number of active sessions is given by $A'(1)$ and equals $S'(1)/(1 - \alpha)$. This means that the mean number of sessions is the product of the mean number of newly started sessions per slot and the mean session length.

The pgf of the steady-state total number of packet arrivals per slot is given by

$$M(x) = A(P(x)). \quad (14)$$

We can also determine the equilibrium condition: the mean number of packet arrivals per slot has to be strictly less than the number of packets that can be transmitted per slot. This implies

$$P'(1)A'(1) < \sigma \Leftrightarrow \frac{P'(1)S'(1)}{1 - \alpha} < \sigma.$$

The equilibrium condition can also be expressed in terms of the mean load ρ of the system:

$$\rho = \frac{P'(1)S'(1)}{\sigma(1 - \alpha)} < 1. \quad (15)$$

4.2. Buffer occupancy

4.2.1. Moments of the buffer occupancy

Let us look at the buffer occupancy at the end of an arbitrary slot in the steady state. First, we demonstrate a technique to calculate the various moments of the buffer occupancy directly from the functional equation (12), i.e.

without first requiring an explicit expression for the pgf $U(z)$ of the buffer occupancy. This method moreover turns out to be less complex than the derivation of the moments based on the pgf $U(z)$.

If we consider those values of x and z , for which the respective arguments of the function Q in the left and right side of the functional equation (12) are equal to each other, we can solve the functional equation. This means that we have to choose x as follows:

$$x = C(xP(z)) = \frac{1 - \alpha}{1 - \alpha P(z)}. \quad (16)$$

Introducing this into the functional equation (12), we get the following expression:

$$Q\left(\frac{1 - \alpha}{1 - \alpha P(z)}, z\right) = \frac{\sigma(z - 1)p_0 S(L(P(z)))}{z - S(L(P(z)))\Phi(z)}. \quad (17)$$

The value of p_0 can be calculated from the normalisation condition $Q(1, 1) = 1$. Putting $z = 1$ in (17), we get that p_0 equals $1 - \rho$, as expected.

Differentiation of both sides of equation (17) with respect to z and evaluation at $z = 1$ yields

$$A'(1)\frac{\alpha P'(1)}{1 - \alpha} + U'(1) = \frac{P'(1)S'(1)[1 - \alpha - P'(1)S'(1) + \alpha P'(1)]}{(1 - \alpha)[\sigma(1 - \alpha) - S'(1)P'(1)]} + \frac{S''(1)P'(1)^2 + S'(1)P''(1)(1 - \alpha)}{2(1 - \alpha)[\sigma(1 - \alpha) - S'(1)P'(1)]},$$

from which we can derive an explicit expression for the mean buffer occupancy $U'(1)$. We rewrite this expression in terms of the variances σ_S^2 and σ_P^2 of the number of new sessions started per slot and the number of packets generated per session per slot respectively:

$$U'(1) = \frac{1}{2(1 - \alpha)^2 \sigma p_0} \{S'(1)(1 - \alpha)\sigma_P^2 + P'(1)^2 \sigma_S^2 + P'(1)S'(1)[1 - \alpha + P'(1)(\alpha - S'(1))]\} - \frac{\alpha P'(1)S'(1)}{(1 - \alpha)^2}. \quad (18)$$

In this expression we clearly see that the mean buffer occupancy is a linearly increasing function of these variances. This has a logical explanation: when the values of the variances σ_S^2 and σ_P^2 increase, the arrival process becomes less predictable and shows more ‘variability’. It is a common result in queueing theory that more ‘variable’ arrival processes lead to higher buffer occupancies.

We can also derive closed-form results for the higher-order moments of the buffer occupancy in a recursive way. The n th-order moment of the buffer occupancy is obtained by differentiating equation (17) n times and evaluating the result for $z = 1$. Unknown partial derivatives in this result are obtained recursively from the functional equation (12).

4.2.2. Pgf of the buffer occupancy

The pgf $U(z)$ of the buffer occupancy can also be computed explicitly: we obtain $U(z)$ by putting x equal to one in the functional equation and by successively applying this equation. In order to determine how the first argument of the Q -function evolves during these successive applications, we first define

$$N_i(z) \triangleq C(P(z)N_{i-1}(z)) = \frac{1 - \alpha + \alpha(1 - P(z))(\alpha P(z))^i}{1 - \alpha P(z)}, \quad (19)$$

and we notice that

$$\lim_{i \rightarrow \infty} N_i(z) = \frac{L(P(z))}{P(z)}, \quad \text{for } |\alpha P(z)| < 1. \quad (20)$$

The recursion formula (19) for $N_i(z)$ and the functional equation (12) then lead to

$$\begin{aligned} U(z) &= Q(1, z) = Q(N_0(z), z) \\ &= \frac{S(P(z)N_0(z))}{z} \{ \Phi(z)Q(N_1(z), z) + \sigma(z-1)p_0 \} \\ &= \dots \\ &= \sigma(z-1)p_0 \left(\sum_{i=0}^{\infty} \frac{1}{\Phi(z)} \prod_{j=0}^i \frac{S(P(z)N_j(z))\Phi(z)}{z} \right) \\ &\quad + \left(\prod_{i=0}^{\infty} \frac{S(P(z)N_i(z))\Phi(z)}{z} \right) Q\left(\frac{1-\alpha}{1-\alpha P(z)}, z\right). \end{aligned}$$

The quantity $Q(\dots)$ in the right-hand side is given by equation (17), and therefore we get the following explicit expression for the pgf of the buffer occupancy:

$$\begin{aligned} U(z) &= \frac{\sigma(z-1)p_0}{z - S(L(P(z)))\Phi(z)} \left\{ S(L(P(z))) \left(\prod_{i=0}^{\infty} \frac{S(P(z)N_i(z))\Phi(z)}{z} \right) \right. \\ &\quad \left. + [z - S(L(P(z)))\Phi(z)] \left(\sum_{i=0}^{\infty} \frac{1}{\Phi(z)} \prod_{j=0}^i \frac{S(P(z)N_j(z))\Phi(z)}{z} \right) \right\}. \quad (21) \end{aligned}$$

As this explicit expression includes infinite sums and products, the derivation of the various moments of the buffer occupancy from it is more complicated than the method presented in the previous section. We can however use this expression to obtain the tail behaviour of the buffer occupancy, as we will explain next.

4.2.3. Tail behaviour of the buffer occupancy

In practice, we are mainly interested in the probability that the buffer occupancy exceeds a certain (and high) threshold value. As indicated in a.o. [5, 23, 25, 26, 38], this probability can be used to estimate the buffer overflow

probability in a finite-capacity buffer system. In principle, the probability that the buffer occupancy exceeds a given threshold can be calculated based on the pgf $U(z)$ by means of inverse z -transformation techniques [18]. Since we are interested in high threshold values with small probabilities, the exact calculation becomes time consuming and error prone. In order to keep the calculation time small, we introduce an approximation method described in [8, 38]. The probability mass function $u(n)$ is dominated by the contribution of the pole z_0 of the pgf $U(z)$ with the smallest modulus, and can be approximated by

$$u(n) \approx -\frac{\theta_0}{z_0} \left(\frac{1}{z_0}\right)^n, \quad (22)$$

for n sufficiently large, and where θ_0 denotes the residue of $U(z)$ in the point $z = z_0$. As such, the probability of exceeding a threshold of T packets in the buffer is given by the following formula, for sufficiently large values of T :

$$\text{Prob}[u > T] \approx -\frac{\theta_0}{z_0 - 1} \left(\frac{1}{z_0}\right)^{T+1}. \quad (23)$$

How do we identify z_0 and θ_0 ? It can be shown (as in [8]) that the denominator of $U(z)$ (i.e. $z - S(L(P(z)))\Phi(z)$) has exactly one real positive zero outside the unit circle in the complex z -plane with multiplicity one. Therefore, to calculate z_0 , we need to calculate the root of $z - S(L(P(z)))\Phi(z) = 0$ different from 1. It can be calculated numerically with, for instance, the Newton-Raphson algorithm.

The residue θ_0 can now be obtained as follows:

$$\theta_0 = \lim_{z \rightarrow z_0} (z - z_0)U(z) = \frac{\sigma z_0 (z_0 - 1) p_0 \left(\prod_{i=0}^{\infty} \frac{S(P(z_0)N_i(z_0))\Phi(z_0)}{z_0} \right)}{\sigma - S'(L(P(z_0))) L'(P(z_0)) P'(z_0) \Phi(z_0)^2}. \quad (24)$$

Notice the infinite product in this expression. We know however that the factor $S(P(z_0)N_i(z_0))$ goes to $S(L(P(z_0)))$ when i approaches infinity (see equation (20)). This is, due to the definition of z_0 , equal to $z_0/\Phi(z_0)$, so we see that the factors of the infinite product go to 1, as i goes to infinity. We can therefore compute the residue θ_0 up to any desired precision by taking the product over a sufficiently large number of factors:

$$\prod_{i=0}^{\infty} \frac{S(P(z_0)N_i(z_0))\Phi(z_0)}{z_0} \approx \prod_{i=0}^J \frac{S(P(z_0)N_i(z_0))\Phi(z_0)}{z_0}. \quad (25)$$

4.3. Packet delay

We now study the delay experienced by individual packets. The delay of a packet is defined as the time period between the end of the slot in which the packet arrives and the end of the last slot of the packet's transmission time. We assume that the queueing discipline is FCFS (first-come first-served) for packets, and that all packets entering during a certain slot arrive in a random

order, irrespective of what session they belong to. To calculate the packet delay, we consider a tagged packet P that arrives in the buffer during a slot in the steady state. Let u_p denote the buffer occupancy after the arrival slot of P , a_p the number of active sessions during this slot, m_p the total number of packets that arrive in the same slot and q the number of packets that arrive after P in the same slot. Then the packet delay of P is given by

$$d = \sum_{i=1}^{u_p - q} t_i, \quad (26)$$

where t_i denotes the (remaining) transmission time of the i th packet that is present in the buffer after the arrival slot of P and has to be transmitted no later than P . The transmission time is geometrically distributed, with pgf $T(z) = \sigma z / (1 - (1 - \sigma)z)$. Due to the memoryless property of the geometric distribution, the remaining transmission time of the first packet in this sum is independent of the already completed part of the transmission when P arrives.

In order to derive the pgf $D(z)$ of d , we first define $V(x, y, z)$ as the joint pgf of the variables m_k , a_k and u_k for a random slot k in the steady state. These variables are respectively the total number of packet arrivals in slot k , the number of active sessions in slot k and the buffer occupancy after slot k . Based on the system equations of section 3, the steady-state pgf $V(x, y, z)$ can be derived as

$$\begin{aligned} V(x, y, z) &\triangleq \lim_{k \rightarrow \infty} E[x^{m_k} y^{a_k} z^{u_k}] \\ &= \frac{S(yP(xz))}{z} \{ \Phi(z) Q(C(yP(xz)), z) + \sigma(z-1)p_0 \}. \end{aligned} \quad (27)$$

It can be shown, using similar methods as in [7, 9], that the pgf $W(x, y, z)$ of the variables m_p , a_p and u_p is given by

$$W(x, y, z) \triangleq E[x^{m_p} y^{a_p} z^{u_p}] = \frac{x}{M'(1)} \frac{\partial}{\partial x} V(x, y, z). \quad (28)$$

To determine the pgf of the packet delay we need the joint pgf of q and u_p . Because we assume all packet arrivals during a certain slot have a random order, the random variable q has a uniform distribution, only dependent on the total number of packet arrivals in the considered slot. This means that $\text{Prob}[q = \ell | m_p = m] = 1/m$, for $0 \leq \ell \leq m - 1$. Using this, we can derive $B(x, z)$, the joint pgf of q and u_p :

$$\begin{aligned} B(x, z) &\triangleq E[x^q z^{u_p}] = \sum_{j=1}^{\infty} \sum_{i=j}^{\infty} \sum_{m=j}^i \sum_{\ell=0}^{m-1} \text{Prob}[a_p = j, u_p = i, m_p = m] \frac{x^\ell z^i}{m} \\ &= \frac{1}{x-1} \int_{t=1}^x \frac{W(t, 1, z)}{t} dt = \frac{V(x, 1, z) - V(1, 1, z)}{M'(1)(x-1)}. \end{aligned} \quad (29)$$

From the joint pgf of q and u_p , we can derive the pgf $D(z)$ of the packet delay:

$$\begin{aligned} D(z) &= E\left[z^{\sum_{i=1}^{u_p-q} t_i}\right] = E[T(z)^{u_p-q}] \\ &= B\left(\frac{1}{T(z)}, T(z)\right) = \frac{\sigma}{P'(1)A'(1)} [U(T(z)) - p_0], \end{aligned} \quad (30)$$

where we used the fact that $M'(1) = P'(1)A'(1)$ and $U(z) = Q(1, z)$. We obtain an expression for $D(z)$ in terms of the pgf of the buffer occupancy $U(z)$, which we know explicitly. Therefore, the pgf of the packet delay is fully determined and all moments of the packet delay can be expressed in terms of corresponding moments of the buffer occupancy. This result is in conformance with the general result obtained for a discrete-time G/Geom/c queue in [19]. It is also in accordance with Little's law [27].

4.4. Session delay

It is also useful to study the delay of a whole session, especially in the case where we consider a non-interrupted data flow to be one session. The session delay d_s is defined as the time period between the end of the slot in which the first packet of a session arrives and the end of the last transmission slot of the last packet of the session. So, in the case of a file transfer, the session delay represents the time it takes to download the whole file. Determining the pgf of the session delay is complicated, and we therefore only derive the mean value of the session delay. The mean session delay is given by

$$E[d_s] = \sum_{n=1}^{\infty} E[d_{s|n}] \text{Prob}[\text{session lasts } n \text{ slots}], \quad (31)$$

where $d_{s|n}$ denotes the session delay in the case that the session lasts n slots. The probability that a session lasts n slots is given by the probability mass function of the session length $\ell(n)$ (see equation (3)). We need to make a distinction between $d_{s|1}$ and $d_{s|n}$ for $n > 1$.

4.4.1. Delay of a session with length 1

The delay of a session with length 1 is given by the time needed to transmit all the packets in the buffer after the session's arrival slot except the ones that arrived after the last packet of the session. If we define q_s as the total number of packets that arrive in the same slot as the session, but after the last packet of the session, and u_s as the buffer occupancy after the session's arrival slot, we get the following expression for the mean session delay:

$$E[d_{s|1}] = (E[u_s] - E[q_s])E[t]. \quad (32)$$

The mean transmission time $E[t]$ is given by $1/\sigma$. Note that u_s is the buffer occupancy after the arrival slot of a tagged newly started session. Let s_s and a_s denote the total number of new sessions and the number of active sessions

respectively, during the arrival slot of the tagged session. In order to derive the joint pgf of s_s , a_s and u_s for a slot in the steady state where a new session starts, we first need to define $H(x, y, z)$ as the joint pgf of the random variables s_k , a_k and u_k for a random slot k in the steady state. Applying the system equations of section 3, we obtain for the steady-state pgf $H(x, y, z)$:

$$\begin{aligned} H(x, y, z) &\triangleq \lim_{k \rightarrow \infty} E[x^{s_k} y^{a_k} z^{u_k}] \\ &= \frac{S(xyP(z))}{z} \{ \Phi(z)Q(C(yP(z)), z) + \sigma(z-1)p_0 \}, \end{aligned} \quad (33)$$

where we have also used the property that $H(1, y, z) = Q(y, z)$. The joint pgf $G(x, y, z)$ of the random variables s_s , a_s and u_s for a slot in the steady state where a tagged new session starts, is then given by

$$\begin{aligned} G(x, y, z) &\triangleq E[x^{s_s} y^{a_s} z^{u_s}] = \frac{x}{S'(1)} \frac{\partial}{\partial x} H(x, y, z) \\ &= \frac{xyP(z)S'(xyP(z))Q(y, z)}{S'(1)S(yP(z))}, \end{aligned} \quad (34)$$

which can again be shown using similar methods as in [7, 9]. This expression enables us to calculate the mean buffer occupancy after the arrival slot of a tagged new session and the mean number of active sessions during such a slot:

$$E[u_s] = \frac{\partial}{\partial z} G(1, 1, 1) = U'(1) + P'(1)(1 - S'(1)) + \frac{S''(1)P'(1)}{S'(1)}, \quad (35)$$

$$E[a_s] = \frac{\partial}{\partial y} G(1, 1, 1) = A'(1) + (1 - S'(1)) + \frac{S''(1)}{S'(1)}. \quad (36)$$

The only unknown quantity left to derive in equation (32) is $E[q_s]$. If we define m_s as the total number of packet arrivals during the tagged slot and m_t as the number of packets sent by the tagged session, q_s is only dependent on m_s and m_t , due to the random order of the packet arrivals in a certain slot. Specifically, it can be shown that $\text{Prob}[q_s = \ell | m_s = m, m_t = m^*] = \binom{m-\ell-1}{m^*-1} / \binom{m}{m^*}$, for $0 \leq \ell \leq m - m^*$. Therefore, $E[q_s]$ is given by

$$\begin{aligned} E[q_s] &= \sum_{j=1}^{\infty} \sum_{m=j}^{\infty} \sum_{m^*=1}^{m-j+1} \sum_{\ell=0}^{m-m^*} \ell \text{Prob}[a_s = j, q_s = \ell, m_s = m, m_t = m^*] \\ &= \sum_{j=1}^{\infty} E \left[\frac{m_s - m_t}{m_t + 1} \middle| a_s = j \right] \text{Prob}[a_s = j], \end{aligned} \quad (37)$$

where we have used the known property that $\sum_{t=k}^n \binom{t}{k} = \binom{n+1}{k+1}$. To determine the conditional expected value in (37), we need the joint pgf of the random variables m_s and m_t conditioned on a_s . We define this pgf as $\Omega_j(x, y)$, and we find

$$\begin{aligned} \Omega_j(x, y) &\triangleq E[x^{m_t} y^{m_s} | a_s = j] \\ &= E \left[x^{m_t} y^{m_t + \sum_{i=1}^{j-1} p_i} \middle| a_s = j \right] = P(xy)P(y)^{j-1}, \end{aligned} \quad (38)$$

because the number of packets p_i generated per session per slot is independent of the number of sessions and the number of packets generated in another session. Using $\Omega_j(x, y)$, we then obtain

$$\begin{aligned} E\left[\frac{m_s - m_t}{m_t + 1} \mid a_s = j\right] &= \left(\frac{\partial}{\partial y} \int_0^1 \Omega_j(x, y) dx\right) \Big|_{y=1} - 1 + \int_0^1 \Omega_j(x, 1) dx \\ &= (j - 1)P'(1) \int_0^1 P(x) dx. \end{aligned} \quad (39)$$

Finally, combining (32)-(39), we get for the mean session delay of a session with length one:

$$\begin{aligned} E[d_{s|1}] &= \frac{1}{\sigma} \left\{ U'(1) + P'(1)(1 - S'(1)) + \frac{S''(1)P'(1)}{S'(1)} \right. \\ &\quad \left. - \left[\frac{\alpha S'(1)^2 + S''(1)(1 - \alpha)}{(1 - \alpha)S'(1)} \right] P'(1) \int_0^1 P(x) dx \right\}. \end{aligned} \quad (40)$$

4.4.2. Delay of a session with length larger than 1

The session delay of a session that lasts n slots ($n > 1$) is given by the total remaining transmission time needed to send the packets present in the buffer after the first slot, the new packets arriving in the following $n - 2$ slots, and the packets arriving no later than the last packet of the session in the last slot (including the last packet). So if we tag a new session and the slot it started in as slot J , we define $u_{(J+i)}$ as the buffer occupancy after the $(i + 1)$ th slot of the session ($0 \leq i \leq n - 1$), $m_{(J+i)}$ as the total number of packets arriving in the $(i + 1)$ th slot and \bar{q} as the number of packets that do not arrive after the last packet of the session in the last slot. We can then write the mean session delay for a session with length n as:

$$E[d_{s|n}] = \left(E[u_{(J)}] + \sum_{i=1}^{n-2} E[m_{(J+i)}] + E[\bar{q}] \right) E[t]. \quad (41)$$

The first term $E[u_{(J)}]$ is equal to $E[u_s]$ defined in the previous section, and therefore given by equation (35).

The terms $E[m_{(J+i)}]$ are equal to $P'(1)E[a_{(J+i)}]$, where $a_{(J+i)}$ is defined as the number of active sessions in slot $J + i$. The following holds ($1 \leq i \leq n - 1$):

$$a_{(J+i)} = s_{(J+i)} + \sum_{k=1}^{a_{(J+i-1)}-1} c_{(J+i)}^k + 1, \quad (42)$$

as the number of active sessions is equal to the sum of the new sessions and the ones that continue, and where the term 1 represents the tagged session that continues with certainty. By taking the mean of this expression, we find (taking

into account that $E[s_{(J+i)}] = S'(1)$ and $E[c_{(J+i)}^k] = \alpha$:

$$\begin{aligned} E[a_{(J+i)}] - 1 &= S'(1) + (E[a_{(J+i-1)}] - 1)\alpha \\ &= \sum_{k=0}^{i-1} \alpha^k S'(1) + \alpha^i (E[a_{(J)}] - 1). \end{aligned} \quad (43)$$

Here, the term $E[a_{(J)}]$ is equal to $E[a_s]$ defined in the previous section and given by equation (36). Substituting this, we get for $E[a_{(J+i)}]$:

$$E[a_{(J+i)}] = 1 + \frac{S'(1)}{1-\alpha} + \alpha^i \left(\frac{S''(1)}{S'(1)} - S'(1) \right), \quad 1 \leq i \leq n-1. \quad (44)$$

To derive the mean of \bar{q} , we define m_t as the number of packets that the tagged session generates in its last slot. Due to the random order of the packet arrivals in a certain slot, \bar{q} is only dependent on m_t and the total number of packet arrivals during the last slot $m_{(J+n-1)}$. Specifically, it can be shown that $\text{Prob}[\bar{q} = \ell | m_{(J+n-1)} = m, m_t = m^*] = \binom{\ell-1}{m^*-1} / \binom{m}{m^*}$, for $0 \leq \ell \leq m - m^*$. In an analogous way as in the previous section, we get

$$E[\bar{q}] = \sum_{j=1}^{\infty} E \left[\frac{m_t(m_{(J+n-1)} + 1)}{m_t + 1} | a_{(J+n-1)} = j \right] \text{Prob}[a_{(J+n-1)} = j]. \quad (45)$$

Because the random variables $m_{(J+n-1)}$ and m_s are identically distributed, we can use the result in (39) to obtain

$$E \left[\frac{m_t(m_{(J+n-1)} + 1)}{m_t + 1} | a_{(J+n-1)} = j \right] = jP'(1) - (j-1)P'(1) \int_0^1 P(x)dx. \quad (46)$$

Combining (35), (36) and (41)-(46), we obtain for $E[d_{s|n}]$:

$$\begin{aligned} E[d_{s|n}] &= \frac{1}{\sigma} \left\{ U'(1) + P'(1)(1 - S'(1)) + \frac{S''(1)P'(1)}{S'(1)} \right. \\ &\quad + (n-1)P'(1) \left(1 + \frac{S'(1)}{1-\alpha} \right) + P'(1) \frac{\alpha - \alpha^n}{1-\alpha} \left(\frac{S''(1)}{S'(1)} - S'(1) \right) \\ &\quad \left. - \left[\frac{S'(1)}{1-\alpha} + \alpha^{n-1} \left(\frac{S''(1)}{S'(1)} - S'(1) \right) \right] P'(1) \int_0^1 P(x)dx \right\}. \end{aligned} \quad (47)$$

4.4.3. Mean session delay

With the above results, we can derive an explicit expression for the mean session delay:

$$\begin{aligned} E[d_s] &= \sum_{n=1}^{\infty} E[d_{s|n}] (1-\alpha)\alpha^{n-1} \\ &= \frac{1}{\sigma} \left\{ U'(1) + \frac{P'(1)}{1-\alpha} + \frac{P'(1)S''(1)}{S'(1)(1-\alpha^2)} + \frac{P'(1)S'(1)(\alpha^2 + 2\alpha - 1)}{(1-\alpha^2)(1-\alpha)} \right. \\ &\quad \left. - \left[\frac{2\alpha S'(1)}{1-\alpha^2} + \frac{S''(1)}{S'(1)(1+\alpha)} \right] P'(1) \int_0^1 P(x)dx \right\}. \end{aligned} \quad (48)$$

5. Some results

In this section we study the influence of the various parameters involved in the model.

5.1. Mean buffer occupancy

First of all, we look at the mean buffer occupancy for an increasing system load ρ (defined in (15)). We start by focusing on the influence of the mean number of packets $P'(1)$ generated by a session in a slot. We assume that the packet generating process is given by the pgf $P(z) = z^p$: an active session generates a constant number of p packets per slot, and the variance σ_P^2 equals zero. The number of new sessions started per slot is a Bernoulli variable and is either 0 or 1 with probabilities $1 - \beta$ and β : its pgf $S(z)$ is given by $1 - \beta + \beta z$, and its variance σ_S^2 is equal to $\beta(1 - \beta)$.

In figure 2(a) we look at the mean buffer occupancy for different values of $P'(1) = p$, by varying the parameter σ of the transmission time to obtain an increasing load. The mean session length $T_s = 1/(1 - \alpha)$ and β (and therefore also the variance σ_S^2) are kept constant. This means that, when we look at a certain value of the system load, we maintain the same load for increasing values of p by proportionally increasing σ , as the load is given by $\beta p T_s / \sigma$. So, when we increase the number of packets sent per slot, we maintain a constant load by decreasing the transmission time of each packet. However, we see in figure 2(a) that the mean buffer occupancy increases when we consider higher values of p . This can be intuitively understood as follows: when we look at a constant value of the load, the mean total transmission time to remove a batch of p packets is the same for all values of p , because the transmission time decreases proportionally for increasing values of p . But, for higher values of p , more packets arrive simultaneously in the system, giving rise to higher values of the buffer occupancy during the time needed to remove these packets. This results in a higher value of the mean buffer occupancy. So, if we consider the situation of a file server where one session equals the download of one file, this means that there will be a lower mean buffer occupancy (and hence a smaller required buffer size) for lower values of p . The value of p denotes the difference between the bandwidth of the input channel and the bandwidth of the output channel: the input bandwidth is p times higher than the output bandwidth.

We can also study the mean buffer occupancy for different values of σ , while varying the mean session length T_s . The result is shown in figure 2(b). When we consider a certain value of the system load, we need an increasing value of the session length T_s to compensate for higher values of σ (and hence shorter transmission lengths). So, for a fixed load, the mean total transmission time needed to remove all packets of a session is the same for all values of σ . However, we notice in figure 2(b) that the mean buffer occupancy increases when σ increases. When σ increases, the mean duration of the sessions is longer and more packets arrive shortly after each other in the system. So, even though the mean transmission time needed to remove all these packets is kept constant, in the mean time higher values of the buffer occupancy are reached, which leads

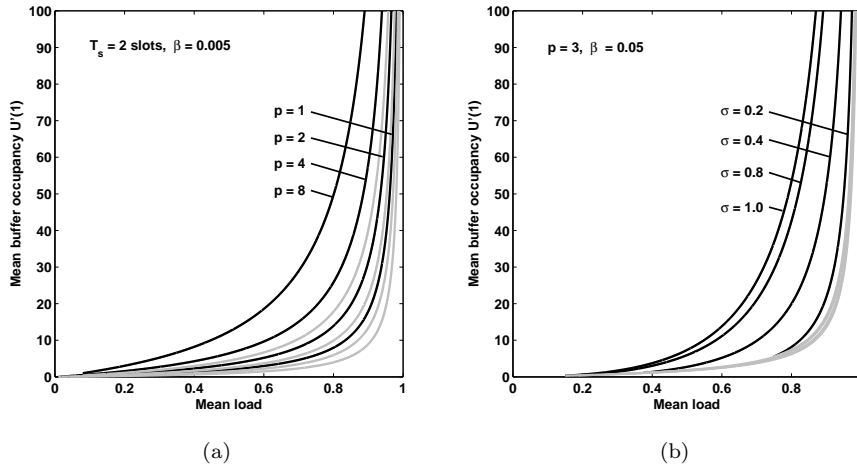
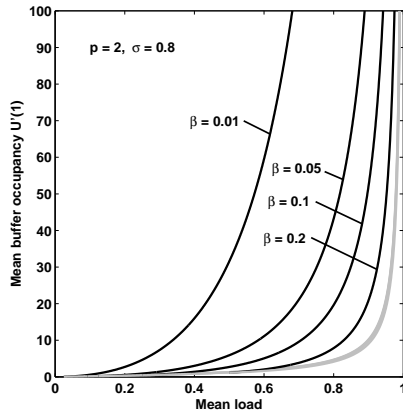


Figure 2: Mean buffer occupancy versus the load ρ for different p by varying the parameter σ of the transmission time (a), and for different σ by varying the mean session length T_s (b) (Grey line = uncorrelated arrivals).

to a higher mean buffer occupancy. If we look again at the example of the file server, this means that for longer mean session lengths (and therefore larger file downloads), the mean buffer occupancy will increase even when the mean transmission time per byte decreases proportionally.

In figure 3 the mean buffer occupancy is shown for different values of the probability β that a new session starts. A table with corresponding numerical results is also included. The distribution of the transmission time (i.e. the parameter σ) and the number of packets p generated per session per slot are kept constant. This means that, when looking at a certain value of the load ρ , an increasing value of β has to be compensated by a proportionally decreasing mean session length T_s . We see that for a higher value of β , or put otherwise, for more but shorter sessions the mean buffer occupancy is lower. Again this can be intuitively understood: more but shorter sessions lead to smoother traffic that is more spread in time, whereas fewer but longer sessions lead to more bursty traffic. In the latter case, the buffer occupancy reaches higher values than in the first case, so the mean buffer occupancy will also be higher. So, in the case of a file server: if there are more downloads, but the files are smaller, then the mean buffer occupancy will be smaller than when there are less downloads, but with larger files.

In figures 2 and 3, we have also plotted the mean buffer occupancy in the case of uncorrelated arrivals: the total number of packets generated per slot is determined by the same pgf $M(x)$ as found in equation (14). The arrival process is however uncorrelated: the number of packets arriving in a slot is independent from slot to slot. The mean buffer occupancy for this uncorrelated



ρ	$E[u]$		
	$\beta=0.01$	$\beta=0.05$	$\beta=0.1$
0.1	0.584	0.129	0.072
0.2	2.715	0.539	0.267
0.3	7.329	1.404	0.663
0.4	15.987	3.016	1.395
0.5	31.500	5.900	2.700
0.6	59.490	11.106	5.058
0.7	113.073	21.084	9.585
0.8	231.600	43.184	19.632
0.9	611.820	114.156	51.948

Figure 3: Mean buffer occupancy versus the load ρ for different probabilities β that a new session starts by varying the mean session length T_s (Grey line = uncorrelated arrivals).

arrival process is shown with grey lines. We see that neglecting the correlation leads to a considerable underestimation of the mean buffer occupancy.

The figures 2 and 3 confirm that more bursty traffic leads to a higher mean buffer occupancy, which is a well-known result in queueing theory. More combinations of varying parameters can also be examined, not always leading to results that can be easily intuitively explained. For instance, looking at the mean buffer occupancy for different values of p while varying T_s , gives results that are dependent on the exact value of the probability β that a new session starts. For this comparison, the variance of the number of new sessions $\sigma_S^2 = \beta(1 - \beta)$ plays an important role. If β is kept small (≤ 0.01), higher values of p (more packets per slot, shorter sessions) lead to higher mean buffer occupancies. When the value of β (and hence the variance σ_S^2) is increased, the curves shift until for $\beta \geq 0.1$ the highest mean buffer occupancy is reached for the lowest value of p (less packets per slot, longer sessions).

We study the influence of the variance σ_P^2 by considering different choices for the pgf $P(z)$ of the packet generation per slot per session:

$$P_1(z) = z^p, \quad P_2(z) = \frac{(1 - \theta)^2 z}{(1 - \theta z)^2}, \quad P_3(z) = \frac{(1 - \lambda)z}{1 - \lambda z},$$

i.e., a deterministic distribution, a negative binomial distribution and a geometric distribution, respectively. We choose the parameters of the distributions so that the mean number of packets generated per session per slot is always the same and equal to p . The variances are then given by

$$\sigma_1^2 = 0, \quad \sigma_2^2 = \frac{1}{2}(p - 1)(p + 1), \quad \sigma_3^2 = p(p - 1),$$

and therefore increasing. The number of new sessions is again assumed to be a Bernoulli variable with parameter β . The variation of the load is realised

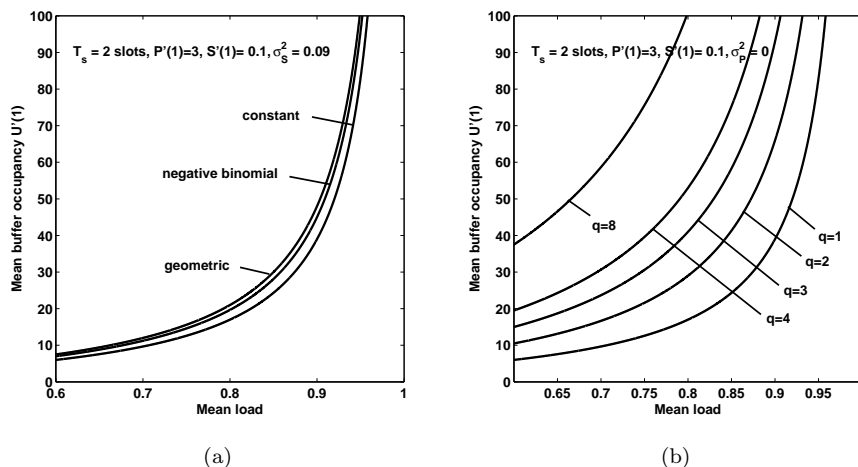


Figure 4: Mean buffer occupancy versus the load ρ for different distributions of the packet generation (a), and for different distributions of the number of new sessions $S(z) = 1 - \gamma + \gamma z^q$ (b).

by varying σ . Because of the stability condition, the load ρ can only range from $P'(1)S'(1)T_s$ to 1. We see in figure 4(a) that the mean buffer occupancy indeed increases for a higher variance of the packet generation. The difference in mean buffer occupancy between the geometric distribution and the negative binomial distribution is however small, even though in the considered situation the variance of the geometric distribution is 1.5 times the variance of the negative binomial distribution.

We can also study the influence of the variance σ_S^2 of the number of new sessions while keeping the mean number of new sessions constant. We assume a constant packet generating process: $P(z) = z^p$ and $\sigma_P^2 = 0$. The pgf of the number of new sessions is given by $1 - \gamma + \gamma z^q$. The mean number of new sessions is then given by $S'(1) = \gamma q$ and the variance by $\sigma_S^2 = \gamma q^2(1 - \gamma) = S'(1)(q - S'(1))$. If we let the value of q increase and adjust the value of γ so that $S'(1)$ is always the same, the variance will linearly increase for higher values of q . In figure 4(b) we see the mean buffer occupancy for different values of q . Again, the variation of the load is brought along by varying σ . All other parameters are chosen the same as in the previous case. We clearly see that a higher variance of the number of new sessions leads to a considerably higher mean buffer occupancy, and that the variance σ_S^2 of the number of new sessions has a greater influence than the variance σ_P^2 of the packet generation process. Note that the curve for constant packet generation in figure 4(a) is the same as the curve for $q = 1$ in figure 4(b).

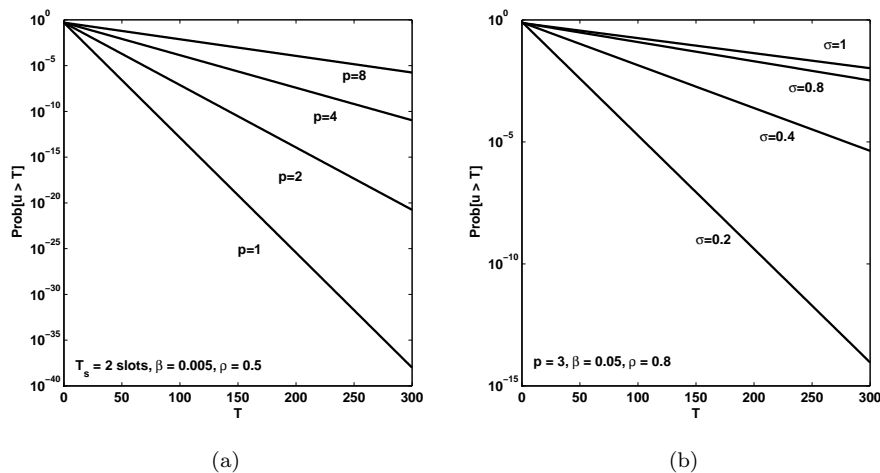


Figure 5: The tail distribution $\text{Prob}[u > T]$ of the buffer occupancy for a constant load, for different p (a), and for different σ (b).

5.2. Tail behaviour of the buffer occupancy

In figure 5 the tail behaviour of the buffer occupancy is plotted under the same conditions as in figure 2. So, in figure 5(a) we keep the mean session length T_s , the probability β of a new session and the load ρ fixed, and we consider the probability that the buffer occupancy exceeds a certain threshold value T for different values of p (the number of packets generated per slot per session). From figure 2(a) it was seen that the lowest mean buffer occupancy was reached for the smallest values of p , and we can make the same observation here: the probability that a certain threshold T is exceeded diminishes for smaller values of p . We also notice that the difference between the different curves is considerable. For $p = 8$, the probability that the buffer occupancy exceeds 300 packets is approximately $1.8 \cdot 10^{-6}$, while for $p = 1$ this probability is equal to $9.9 \cdot 10^{-39}$: this is a significant difference, taking into account that the load is the same in both cases.

In figure 5(b) the tail behaviour of the buffer occupancy is depicted under the same conditions as in figure 2(b). We already concluded from figure 2(b) that the mean buffer occupancy increases for higher values of σ and the same can be observed here for the tail behaviour of the buffer occupancy. When σ increases and T_s increases accordingly maintaining a fixed load, the probability that the buffer occupancy exceeds a certain threshold T also grows. Notice also that the tail of the buffer occupancy for $\sigma = 0.8$ and $\sigma = 1$ does not decay very rapidly.

Observing the tail behaviour of the buffer occupancy leads to the same conclusions as studying the mean buffer occupancy. The tail behaviour however allows us to know more about peak values, useful for the dimensioning of a data

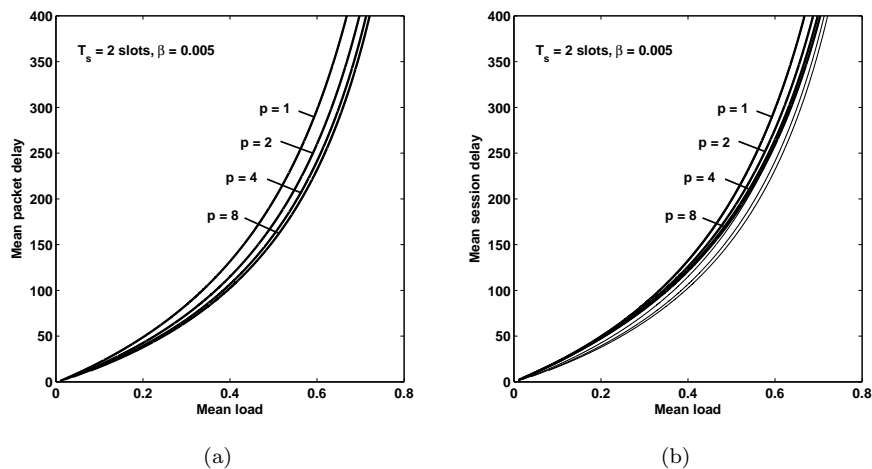


Figure 6: The mean packet delay versus the load ρ for different p (a), and the mean session delay versus the load (b) (Thin line = mean packet delay).

buffer.

5.3. Packet delay and session delay

We reconsider the situation depicted in figure 2(a): the mean session length T_s and the probability β that a new session starts are kept constant. During each slot of a session a fixed number of packets is generated, and this number is given by p . In figure 6(a) we see the mean packet delay for various values of p , and in figure 6(b) the mean session delay is shown and as a comparison, the mean packet delay is also displayed with thin lines. The variation of the load is obtained by adjusting the value of σ (i.e. the length of the transmission times). We see that for an increasing value of the load ρ , and hence an increasing packet transmission time, the delays rise quickly. However, the mean packet delay and the mean session delay for a certain value of p do not show much difference. Especially for p equal to one, both curves are almost identical, the mean session delay being slightly bigger than the packet delay. This can be explained by the fact that the mean session length in the considered case is 2, so most sessions are not very long. However, the mean session delay includes the transmission time of (on average) $2p$ packets, whereas the mean packet delay only includes the transmission time of 1 packet. The fact that the actual difference is smaller than the transmission time needed for $2p - 1$ packets is due to the difference in buffer occupancy after a packet or a new session arrival. Note that the buffer occupancy can never decrease as long as there are active sessions. It is therefore intuitively easily understood that the buffer occupancy after the start of a session is lower than the buffer occupancy after a random packet arrival (which can be at the beginning, in the middle or at the end of a session). During

the time that a session is active, the buffer occupancy will reach higher values than just before it started. The buffer occupancy after a packet arrival is an important factor of the delay of this packet, and the buffer occupancy after a newly started session is an important part of the session delay. So, if the buffer occupancy after a random packet arrival is higher, this surplus compensates some of the $2p - 1$ packets more to be sent for the session delay, leading to a smaller difference in delay for the same values of p . Of course, other factors are also incorporated in the delay, e.g. the transmission times of the packets that arrive after the tagged packet do not contribute to the packet delay, and the transmission times of packets of other sessions are also included in the session delay.

Notice that increasing the mean session length does not substantially change the difference between packet delay and session delay: if a session lasts longer, more packets have to be transmitted, but the buffer occupancies during this session also reach higher values. Therefore, for longer sessions, the session delay increases due to the fact that more packets arrive in a session, but the packet delay increases too due to the higher buffer occupancy after a random packet. Because we maintain a certain load by varying the parameter σ of the transmission times, a longer mean session length T_s will be compensated by shorter transmission times (i.e. a higher value of σ). So, for higher session lengths, the packet or session delay includes the transmission of more packets, but the actual transmission times are shorter. The consequence of this is that the difference between mean packet and mean session delay will always be rather small, irrespective of the session length. Only for $p = 1$ we see that the difference between session and packet delay will increase significantly when the mean session length increases. Because the packets during a session only arrive at the rate of one packet per slot, the buffer occupancy after a random packet arrival will not reach very high values. For higher mean session lengths T_s however, the mean transmission time decreases proportionally (to maintain a certain load): so, for a higher mean session length the mean packet delay decreases, whereas the mean session delay appears to remain the same. This is shown in figure 7(a). In figure 7(b) the mean values of the packet and session delay are shown for $p = 4$: when T_s increases, the difference between mean packet and session delay does not increase either.

If we compare the mean packet delay (and session delay) illustrated in figure 6 to the mean buffer occupancy shown in figure 2(a), we see that the mean packet delay behaves differently than the mean buffer occupancy depending on the value of p . While the mean buffer occupancy increases for an increasing value of the number of packets p , the mean packet delay decreases when p rises. When p rises, the buffer occupancy rises too, so more packets have to be sent before the tagged packet can be transmitted. An increasing buffer occupancy leads to higher packet delays. But on the other hand, when p increases, the parameter σ has to increase to maintain a certain load. So, for a higher number p of packet arrivals, the transmission time needed to send a packet becomes smaller. A smaller transmission time has a positive influence on the packet delay by making it also smaller. This means that for higher values of p we have

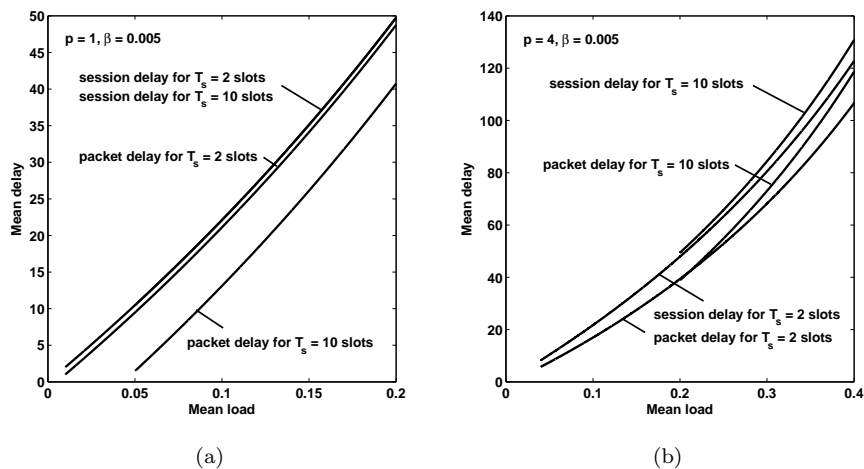


Figure 7: The mean packet and session delay versus the load ρ for two different mean session lengths and $p = 1$ (a), and for $p = 4$ (b).

two effects that counteract: we see in figure 6(a) that the packet delay decreases for higher values of p , and we can conclude that the effect of the smaller transmission time is larger than the effect of the increasing buffer occupancy on the mean packet delay.

We also look at the mean packet delay and the mean session delay in case of the situation considered in figure 2(b): the number of packets p and the probability β that a new session starts are fixed, and we study the delays for various values of σ . The results are shown in figure 8. We see that in this case the mean packet and session delay have the same behaviour as the mean buffer occupancy (figure 2(b)). Notice that there are two opposing effects for a higher value of σ : when σ increases, the transmission time of a packet decreases, so this has a positive effect on the packet (or session) delay. However, for higher values of σ , the mean buffer occupancy is higher, which leads to higher delays. The last effect has apparently the most impact: for higher values of σ (i.e. shorter transmission times) and hence higher mean session lengths, we see in figure 8 that the delay increases. We see however that there is not much distance between the different curves: for $\sigma = 0.8$ and $\sigma = 1.0$, the curves are almost equal to each other. The differences between the corresponding mean buffer occupancies in figure 2(b) are more substantial. We also notice that there is again not much difference between the mean packet delay and the mean session delay. This can be explained in the same way as before.

Finally we study the mean packet delay and the mean session delay when the number of packets p and the mean transmission time (so the parameter σ) are kept constant. This is the situation also depicted in figure 3. The load is varied by adjusting the mean session length T_s and we look at the delays for

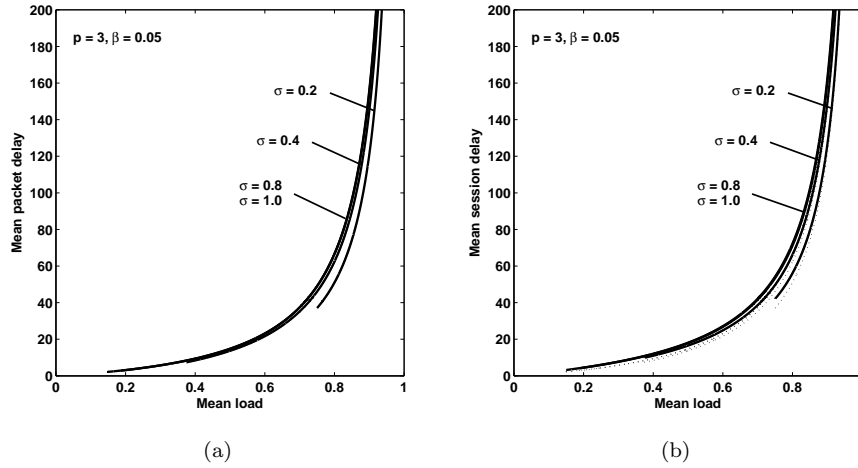


Figure 8: The mean packet delay versus the load ρ for different σ (a), and the mean session delay versus the load (b) (Dashed line = mean packet delay).

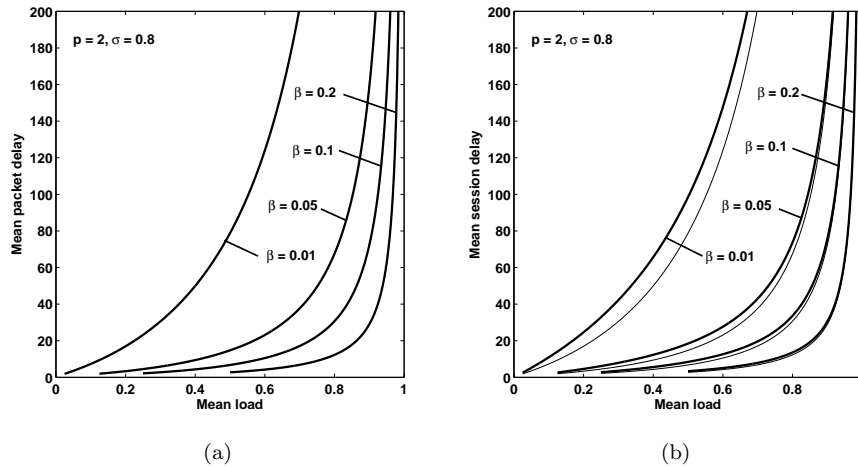


Figure 9: The mean packet delay versus the load ρ for different probabilities β that a new session starts (a), and the mean session delay versus the load (b) (Thin line = mean packet delay).

different values of β (the probability that a new session starts). The results are shown in figure 9. The mean packet and session delay behave in the same way as the corresponding mean buffer occupancy (figure 3). This is a logical result, as we only vary parameters of the arrival process and the mean transmission time of a packet is kept constant. The delay is therefore only determined by the buffer occupancy: if the buffer occupancy rises, the delay will increase too.

We see in figure 9(b) that the mean session delay does not differ much from the mean packet delay for the same value of β . The difference between both delays increases for lower values of β . Since lower values of β correspond with higher values of the mean session length T_s to maintain a certain load (i.e. less but longer sessions instead of more but shorter sessions), this increasing session length leads to a higher difference between the mean session and packet delay. In this situation the distribution of the transmission time is kept constant, so (in contrary to the situation depicted in figure 7) longer session lengths are not compensated by smaller transmission times, explaining the higher difference between mean packet and session delay for longer sessions.

6. Applying the model to a web server

We consider a web server: this is a computer system that accepts requests from clients for a certain web page or embedded file (e.g. a picture or a movie), and that responds by sending the requested file to the client. A web server generally contains the web pages of many web sites. We apply our model on the situation depicted in figure 10: the web server is connected to the internet through a gateway. This gateway contains a data buffer for outgoing data (from the server to the internet). If we consider the transmission of one file to a client as one session, it is obvious that the buffer for outgoing data will experience session-like data traffic. We also suppose that the data transfer from the server to the gateway is uninterrupted. The model proposed in this paper can therefore be applied. Fixed-length packets requiring one slot of transmission time are considered. The gateway has an unreliable output line subject to random failures, occurring independently from slot to slot.

Usually, the HTTP traffic is transported to the user over a reliable connection using the TCP protocol. TCP provides both flow and error control by employing an acknowledgement- and window-based communication between source and destination, see e.g. [1, 29]. As a result of TCP's strategies like AIMD (additive increase/multiplicative decrease) and 'slow start' that are driven by signs of network congestion, the instantaneous transmission rate of the source adapts dynamically over time. The throughput of TCP sessions is studied in e.g. [4]. In our application however, we assume that the network between the web server and the users is relatively free of bottlenecks. Under these conditions, the throughput of the sessions is determined by the speed of the web server rather than by TCP's congestion control. More intricate models for the traffic generated by web servers can be found in a.o. [28, 30, 34].

We need to assign realistic values to the parameters that are involved in the model. First of all, we look at the pgf $P(z)$ of the number of packets

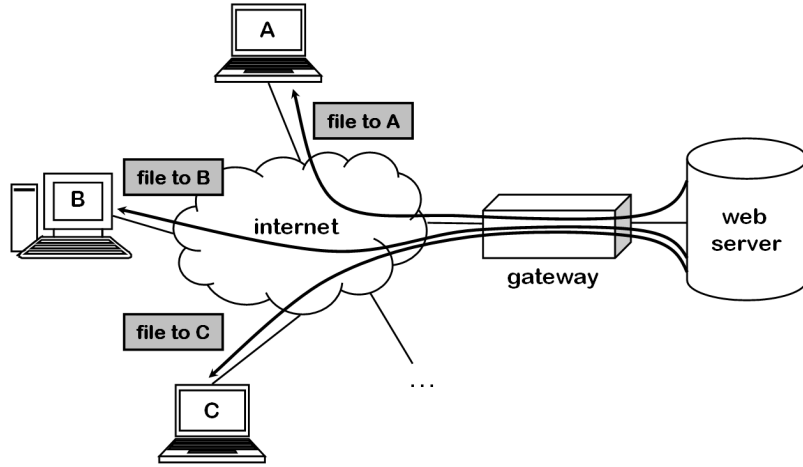


Figure 10: A web server connected to the internet through a gateway.

generated per session per slot. We assume that the output line of the gateway (connected to the internet) has a typical bandwidth of 100 Mbit/s. The pgf $P(z)$ is dependent on the characteristics of the web server itself: we assume that the web server has a maximum transfer rate of 1000 Mbit/s (as is the case for e.g. the HP ProLiant DL585 Rack Storage Server). Therefore, the pgf $P(z)$ can be written as $\sum_{i=1}^I \text{Prob}[p = i] z^i$, where the value of I is 10, as the transfer rate is at most 10 times higher than the bandwidth of the output channel. We need to choose the probabilities $\text{Prob}[p = i]$ in a realistic manner. For a certain number of packets p per slot, the transfer rate is equal to p times 100 Mbit/s. The assignment of the probabilities to these values is done by dividing the values of p into three classes. When $p = 1-3$, the transfer rate per session equals 100–300 Mbit/s, so this corresponds with a low transfer rate per session. Because this low transfer rate occurs the least frequently, we assign a weight factor equal to 1 to these p -values. Analogously, the second class contains $p = 4-6$ (medium transfer rate) and has a weight factor 3, because a medium transfer rate will occur the most frequently. The third class contains $p = 7-10$ (high to maximum transfer rate) and has a weight factor 2. The pgf $P(z)$ then looks as follows:

$$P(z) = \frac{1}{20} (z + z^2 + z^3) + \frac{3}{20} (z^4 + z^5 + z^6) + \frac{2}{20} (z^7 + z^8 + z^9 + z^{10}), \quad (49)$$

and $P'(1) = 5.95$ packets are generated per session per slot. This means that we assume that the mean transfer rate per file from the server equals 5.95 times 100 Mbit/s (so 595 Mbit/s). Note that the generation of packets was assumed to be independent from slot to slot. In reality, this is not the case: the generation of packets is dependent on the number of simultaneous active

sessions (the bandwidth has to be shared between the sessions), and because sessions continue in time, the packet generation is not independent from slot to slot.

Until now, we have not defined the length of a slot (in seconds). If we assume that the parameter σ models the availability of the output channel (see further), the slot length t_{slot} is the time needed to send one packet if the channel is available. We define the size of one packet as 100 bytes. Because the bandwidth of the output channel is 100 Mbit/s, the slot length is given by (100 bytes)/(100 Mbit/s). So, the slot length t_{slot} is equal to 8 μ s.

At <http://ita.ee.lbl.gov/html/contrib/EPA-HTTP.html>, a trace of real web traffic can be found. A trace is a log file that contains all the file requests of clients during a whole day. The trace contains a day's worth of all HTTP requests to the EPA WWW server located at Research Triangle Park, NC. We removed the invalid and empty requests from it, and extracted the time stamp and the byte size of all the file requests. Each file request then corresponds to a session. If we look at the byte sizes of the files occurring in the trace, it is possible to approximate the distribution of the file size by a geometric distribution, for the smallest sizes (a size smaller than 6000 bytes). For higher file sizes, the geometric distribution is a coarser approximation: the distribution of the actual file sizes is heavy-tailed [13]. The files on a web server are typically either small or either very large [3], hence this result. We cut off this heavy tail by neglecting all files with a size greater than 6000 bytes and as a result we obtain that the mean file size (in the trace) is equal to 1862 bytes. To derive a value for the mean session length T_s (and hence the parameter α), we need this mean file size. We know that the pgf of the total number of packets generated by a session is given by

$$P_{tot}(z) = E[z^{p_{tot}}] = E\left[z^{\sum_{k=1}^{\ell_i} p_k^i}\right] = L(P(z)),$$

where $L(z)$ is given by equation (4) and $P(z)$ by equation (49). We match the first moment of this distribution to the mean file size to obtain an equation for the mean session length $T_s = L'(1)$:

$$E[p_{tot}] = L'(1)P'(1) = \frac{E[\text{file size in trace}]}{100 \text{ bytes}}.$$

Solving this equation gives the following result:

$$T_s = 3.13 \text{ slots},$$

and hence

$$\alpha = 0.6805.$$

Left to determine is the pgf $S(z)$ of the number of new sessions per slot. Because the slot length is small, we assume that at most one new session begins per slot, so $S(z) = 1 - \beta + \beta z$. The parameter β is the probability that one new session starts. We look at a web server that serves a lot of customers during a

Table 1: Performance measures of a web server for different values of the output channel availability σ .

	$\sigma = 0.95$	$\sigma = 0.5$	$\sigma = 0.2$
System load ρ	0.157	0.298	0.745
Mean number of active sessions $A'(1)$	0.025	0.025	0.025
Mean buffer occupancy $U'(1)$	2.756	6.696	47.936
Mean packet delay $D'(1)$	18.499 slots = 147.99 μs	44.944 slots = 359.55 μs	321.76 slots = 2574.1 μs
Mean session delay $E[d_s]$	22.723 slots = 181.78 μs	51.053 slots = 408.42 μs	333.84 slots = 2670.68 μs
Mean buffer occupancy after a random packet arrival $E[u_p]$	20.634	25.532	67.412
Mean buffer occupancy after the first slot of a session $E[u_s]$	8.658	12.598	53.839
Prob[$u > 100$]	$0.363 \cdot 10^{-3}$	0.00317	0.156
Prob[$u > 1000$]	$7.35 \cdot 10^{-25}$	$2.41 \cdot 10^{-20}$	$1.24 \cdot 10^{-7}$

rush period. If we assume that on average 1 session is started per millisecond, β is equal to 0.008. Note that, as the number s_k of new sessions in a slot is a Bernoulli variable, there is no correlation between the number of sessions started in different slots. In reality however, this is generally not true: the number of new sessions started will (among other things) depend on the time of the day.

Finally, we have to find an appropriate value for σ . We assume that σ is a measure for the availability of the output channel and we look at the performance measures for three different values of σ : $\sigma = 0.95$ corresponds with the situation where the output line is available 95% of the time, $\sigma = 0.5$ when it is available half of the time, and $\sigma = 0.2$ corresponds with an output line availability of 20% (unreliable channel). The resulting performance measures are given in table 1. A lower value of σ corresponds with a higher load: the load for $\sigma = 0.2$ is almost five times higher than the load for $\sigma = 0.95$. We see that the mean buffer occupancy, the mean packet delay and the mean session delay increase with a factor higher than 5. For $\sigma = 0.2$, the system becomes slightly overloaded: the mean packet delay and the mean session delay have a dimension of milliseconds, while the actual mean transmission time is only $1/\sigma = 40 \mu s$. We also see that the probability that the buffer occupancy exceeds 100 packets (so, at least 10 kbytes of data are stored in the buffer and waiting to be transmitted) is about 1/6. The mean number of active sessions is (logically) the same for all values of σ , as it does not depend on σ .

As this example shows, our model gives us the opportunity of quickly comparing different situations corresponding with different values of the system parameters, for dimensioning purposes.

7. Conclusions and future work

In this paper we have proposed an analytical technique for the performance evaluation of a buffer system with session-based arrival streams. We have de-

rived the characteristics of the arrival process, the buffer occupancy and the packet delay. An expression for the mean session delay was also found. With the expression found for the pgf of the buffer occupancy, we have also derived an approximation for the tail distribution of the buffer occupancy, which is useful for the dimensioning of buffer sizes. Furthermore, it is known that the overflow probability $\text{Prob}[u > T]$ of the buffer occupancy exceeding level T in an infinite-capacity system can be used to estimate the loss probabilities in the corresponding system with finite capacity T with reasonable accuracy.

By adjusting the parameters of the system, we can study their influence. We have applied the model to a web server, based on a trace of actual web traffic. It is possible to fit the data to the model, by introducing some restrictions. The most important restriction is the assumption that the session length is geometric. For the smallest file sizes, this seems like a reasonable assumption. For larger files, the distribution is however heavy-tailed. More accurate results can therefore be obtained by allowing non-geometric session lengths. This will be the subject of future research. One could e.g. allow for session lengths that are a probabilistic mixture of geometrics, i.e.

$$L(z) = \sum_{i=1}^N p_i \frac{(1 - \alpha_i)z}{1 - \alpha_i z} \quad \text{with} \quad \sum_{i=1}^N p_i = 1.$$

In [17] it is shown that a mixture of exponentials can be used to approximate a heavy-tailed file size distribution. This would probably require an additional system state variable for each ‘phase’ of the distribution.

A further logical extension is allowing $P(0)$ to be different from 0. This means that a session can also generate no packets in a slot. This is however more complex than it seems. Allowing $P(0) \neq 0$, makes it possible that an infinite number of sessions are active while the equilibrium condition is still fulfilled. An analysis of this system leads to a model with an infinite number of unknown boundary probabilities.

Another issue is the fact that, if we look at the traffic of a web or file server, sessions usually are TCP sessions. The TCP protocol adjusts the sending rate of the sources in order to avoid congestion in the network. Therefore the sending rate changes over time as result of strategies like AIMD (additive increase/multiplicative decrease) and ‘slow start’ [1]. Therefore, it would be useful to examine a model where the number of packets generated per slot during a session is correlated. The simplest choice (and already reasonable adequate) would be a two-state Markov modulated packet generation process, independently for each session. That would require at least one additional variable in the system state description. To model slow start, we could assume that when a session starts, it is always in the ‘bad’ state (with $P'_{bad}(1) < P'_{good}(1)$).

8. Acknowledgements

The first author gratefully acknowledges the financial support from the Research Foundation - Flanders (FWO-Vlaanderen), Belgium.

References

- [1] M. Allman, V. Paxson, W. Stevens, TCP Congestion Control, RFC 2581, Proposed Standard (1999).
- [2] E. Altman, A. Jean-Marie, The distribution of delays of dispersed messages in an M/M/1 queue, in: Proceedings of IEEE INFOCOM '95 (Boston, 2-6 April 1995) 338-344.
- [3] M. Arlitt, C. Williamson, Internet Web servers: workload characterization and performance implications, IEEE/ACM Transactions on Networking 5 (5) (1997) 631-645.
- [4] F. Baccelli, D.R. McDonald, A stochastic model for the throughput of non-persistent TCP flows, Performance Evaluation 65 (6-7) (2008) 512-530.
- [5] C. Bisdikian, J. Lew, A. Tantawi, On the tail approximation of the blocking probability of single server queues with finite buffer capacity, in: Proceedings of the Second International Conference on Queueing Networks with Finite Capacity (Research Triangle Park, 28-29 May 1992) 267-280.
- [6] H. Bruneel, B.G. Kim, Discrete-Time Models for Communication Systems Including ATM, Kluwer Academic Publishers, Boston, 1993.
- [7] H. Bruneel, Packet delay and queue length for statistical multiplexers with low-speed access lines, Computer Networks and ISDN Systems 25 (12) (1993) 1267-1277.
- [8] H. Bruneel, B. Steyaert, E. Desmet, G. Petit, Analytic derivation of tail probabilities for queue lengths and waiting times in ATM multiserver queues, European Journal of Operational Research 76 (3) (1994) 563-572.
- [9] H. Bruneel, Calculation of message delays and message waiting times in switching elements with slow access lines, IEEE Transactions on Communications 42 (2-4) (1994) 255-259.
- [10] B.D. Choi, D.I. Choi, Y. Lee, D.K. Sung, Priority queueing system with fixed-length packet-train arrivals, IEE Proceedings-Communications 145 (5) (1998) 331-336.
- [11] I. Cidon, A. Khamisy, M. Sidi, On queueing delays of dispersed messages, Queueing Systems 15 (1-4) (1994) 325-345.
- [12] I. Cidon, A. Khamisy, M. Sidi, Delay, jitter and threshold crossing in ATM systems with dispersed messages, Performance Evaluation 29 (2) (1997) 85-104.
- [13] M.E. Crovella, A. Bestavros, Self-similarity in World Wide Web traffic: evidence and possible causes, IEEE/ACM Transactions on Networking 5 (6) (1997) 835-846.

- [14] J. Daigle, Message delays at packet-switching nodes serving multiple classes, *IEEE Transactions on Communications* 38 (4) (1990) 447-455.
- [15] S. De Vuyst, S. Wittevrongel, H. Bruneel, Statistical multiplexing of correlated variable-length packet trains: an analytic performance study, *Journal of the Operational Research Society* 52 (3) (2001) 318-327.
- [16] S. De Vuyst, S. Wittevrongel, H. Bruneel, Mean value and tail distribution of the message delay in statistical multiplexers with correlated train arrivals, *Performance Evaluation* 48 (1-4) (2002) 103-129.
- [17] A. Feldmann, W. Whitt, Fitting mixtures of exponentials to long-tail distributions to analyze network performance models, *Performance Evaluation* 31 (3-4) (1998) 245-279.
- [18] W. Feller, *An Introduction to Probability Theory and its Applications*, Volume I, Wiley, New York, 1968.
- [19] P. Gao, S. Wittevrongel, H. Bruneel, Delay against system contents in discrete-time G/Geom/c queue, *Electronics Letters* 39 (17) (2003) 1290-1292.
- [20] J. He, K. Sohraby, On the Queueing Analysis of Dispersed Periodic Messages, in: *Proceedings of IEEE INFOCOM 2000* (Tel Aviv, 26-30 March 2000) 1046-1054.
- [21] J.J. Hunter, *Mathematical Techniques of Applied Probability*, Volume 2, *Discrete Time Models: Techniques and Applications*, Academic Press, New York, 1983.
- [22] G.U. Hwang, B.D. Choi, Closed-form expressions on the geometric tail behavior of statistical multiplexers with heterogeneous traffic, *IEEE Transactions on Communications* 46 (12) (1998) 1575-1579.
- [23] F. Ishizaki, T. Takine, H. Terada, T. Hasegawa, Loss probability approximation of a statistical multiplexer and its application to call admission control in high-speed networks, in: *Proceedings of IEEE GLOBECOM '95* (Singapore, 14-16 November 1995) 417-421.
- [24] F. Kamoun, Performance analysis of a discrete-time queueing system with a correlated train arrival process, *Performance Evaluation* 63 (4-5) (2006) 315-340.
- [25] J. Keilson, Ergodic queue length distribution for queueing systems with finite capacity, *Journal of the Royal Statistical Society Series B – Statistical Methodology* 28 (1) (1966) 190-201.
- [26] H.S. Kim, N.B. Shroff, On the asymptotic relationship between the overflow probability and the loss ratio, *Advances in Applied Probability* 33 (4) (2001) 836-863.

- [27] L. Kleinrock, *Queueing Systems, Volume I: Theory*, Wiley, New York, 1975.
- [28] Z. Liu, N. Niclausse, C. Jalpa-Villanueva, Traffic model and performance evaluation of Web servers, *Performance Evaluation* 46 (2-3) (2001) 77-100.
- [29] S.H. Low, F. Paganini, J.C. Doyle, Internet congestion control, *IEEE Control Systems Magazine* 22 (1) (2002) 28-43.
- [30] B.A. Mah, An empirical model of HTTP network traffic, in: *Proceedings of IEEE INFOCOM '97* (Kobe, 7-12 April 1997) 592-600.
- [31] M.F. Neuts, Probabilistic modelling requires a certain imagination, in: *Proceedings of ICOTS 3* (Dunedin, 19-24 August 1990) 122-131.
- [32] K. Sahraby, On the theory of general ON-OFF sources with applications in high-speed networks, in: *Proceedings of IEEE INFOCOM '93* (San Francisco, 28 March - 1 April 1993) 401-410.
- [33] H. Takagi, *Queueing Analysis – A Foundation of Performance Evaluation, Volume 3: Discrete-Time Systems*, North-Holland, New York, 1993.
- [34] R.D. van der Mei, R. Hariharan, P.K. Reeser, Web server performance modeling, *Telecommunication Systems* 16 (3-4) (2001) 361-378.
- [35] J. Walraevens, S. Wittevrongel, H. Bruneel, A discrete-time priority queue with train arrivals, *Stochastic Models* 23 (3) (2007) 489-512.
- [36] S. Wittevrongel, H. Bruneel, Correlation effects in ATM queues due to data format conversions, *Performance Evaluation* 32 (1) (1998) 35-56.
- [37] S. Wittevrongel, Discrete-time buffers with variable-length train arrivals, *Electronics Letters* 34 (18) (1998) 1719-1721.
- [38] C. Woodside, E. Ho, Engineering calculation of overflow probabilities in buffers with Markov-interrupted service, *IEEE Transactions on Communications* 35 (12) (1987) 1272-1277.
- [39] Y. Xiong, H. Bruneel, Buffer contents and delay for statistical multiplexers with fixed-length packet-train arrivals, *Performance Evaluation* 17 (1) (1993) 31-42.
- [40] Y. Xiong, H. Bruneel, Buffer behaviour of statistical multiplexers with correlated train arrivals, *International Journal of Electronics and Communications (AEÜ)* 51 (3) (1997) 178-186.