Int. J. Appl. Math. Comput. Sci., 2014, Vol. 24, No. 2, 429–442
DOI: 10.2478/amcs-2014-0033

TRANSIENT AND STATIONARY CHARACTERISTICS OF A PACKET BUFFER MODELLED AS AN *MAP/SM/1/B* SYSTEM

KRZYSZTOF RUSEK, LUCJAN JANOWSKI, ZDZISŁAW PAPIR

Department of Telecommunications
AGH University of Science and Technology, al. A. Mickiewicza 30, 30-059 Cracow, Poland
e-mail: krusek@agh.edu.pl, {janowski,papir}@kt.agh.edu.pl

A packet buffer limited to a fixed number of packets (regardless of their lengths) is considered. The buffer is described as a finite FIFO queuing system fed by a Markovian Arrival Process (MAP) with service times forming a Semi-Markov (SM) process ($MAP/SM/1/b$ in Kendall's notation). Such assumptions allow us to obtain new analytical results for the queuing characteristics of the buffer. In the paper, the following are considered: the time to fill the buffer, the local loss intensity, the loss ratio, and the total number of losses in a given time interval. Predictions of the proposed model are much closer to the trace-driven simulation results compared with the prediction of the $MAP/G/1/b$ model.

Keywords: router interface, Markovian arrival process, semi-Markov service time, hidden Markov model, finite-buffer queue, packet loss, first passage time.

1. Introduction

Since the beginnings of computer networks, many models have been developed for interfaces of network devices such as routers. A majority of them were designed for ATM devices and are not suitable for IP devices, because IP packets are not of constant length as opposed to those in ATM/are. Thus, the existing models have to be modified to fit the real IP network more precisely.

In the commonly used models, the router interface is just a piece of memory drained at a constant rate of C bytes per second. This simple model is correct for some types of interfaces, but not for all. In some types of devices (e.g., Cisco ISR routers) the buffer can hold no more than a fixed number of packets, regardless of their lengths. We call this type of buffer *packet-oriented*, and a standard buffer *byte-oriented*.

For a packet-oriented buffer, a packet service is completed when the entire packet is removed from the queue. In a byte-oriented buffer, on the other hand, only part of a packet is removed upon service completion. The main modeling problem for real devices addressed in this paper is that they have packet-oriented buffers which are drained at a constant rate in bytes per second, not packets per second. This problem is discussed in Section 4, while an extended discussion of real packet buffers, their types and properties can be found in the works of Rusek *et al.* (2011), Janowski and Owezarski (2010) or Sequeira *et al.* (2012).

A packet stored in a packet-oriented queue is removed from the buffer after it has been sent to the destination. Since the time required to complete this operation depends on the actual packet length (the link speed is constant), the service process model has to reflect all the statistical properties of packet lengths, including their correlation.

IP network traffic has a rich correlation structure. Packet interarrival times are highly autocorrelated (Paxson and Floyd, 1995). Similar behaviour was observed for packet lengths (Emmert *et al.*, 2007). Finally, packet lengths can be correlated with interarrival times (Salvador *et al.*, 2004). All these correlations will considerably affect the accuracy of any switching device model.

Among many attempts to model the correlation structure of IP traffic with respect to its queuing, Markovian models have proven to be a powerful tool. Despite the fact that they cannot accurately model selfsimilar and long range dependent behaviour over all time scales, they can approximate them up to an arbitrary time scale (Robert and LeBoudec, 1996), which is often sufficient. More importantly, however, in most cases Markovian traffic models lead to analytically solvable queuing models even for complicated service disciplines (Chydziński and Chróst, 2011). This is why they are still attractive and actively explored, (see e.g., Dudin *et al.*, 2005; Chydziński, 2007; 2006; 2008), although a lot of theoretical work was done at the end of the previous century.

The most flexible traffic model from the Markov family is the Batch Markovian Arrival Process (BMAP). This model can handle packet lengths and all the correlations observed in real traffic. It is also possible to construct an analytically solvable queue model with the BMAP at the input (Dudin *et al.*, 2002; Chydziński, 2006). However, all of the models implicitly assume that buffers are byte-oriented. To the authors' best knowledge, a model of a finite packet-oriented buffer fed by the BMAP has not been derived yet. This is not surprising because this system is quite complicated. However, if we assume packet lengths being independent of interarrival times, the model is simplified because the autocorrelation of packet lengths can be represented by the service time autocorrelation and the arrival process is reduced to the MAP while retaining the interarrival times autocorrelation.

The Semi-Markov (SM) process, being a direct generalization of the Markov process (Janssen and Manca, 2006), is often used as a model of the correlated service process in a queue. Therefore, this type of service process was used in the proposed model. Queues with semi-Markov service times have been studied for many years (Neuts, 1966). Most of the theoretical results were obtained for an infinite capacity system. Recently, Dudin *et al.* (2002) explored the general finite system $BMAP/SM/1/b$, using the imbedded Markov chain approach. However, in contrast to our model, none of the analytical results provides the possibility of a *transient state analysis of a finite capacity system*. In most cases, stationary analysis is sufficient. However, transient state analysis reveals more details about the system (Kobayashi and Ren, 1992; Schwefe *et al.*, 2001; Lucantoni *et al.*, 1994).

The loss process in a general $MAP/G/1/b$ system was analyzed by Chydziński (2008) using a novel approach called the *potential method*. In this method the system of integral equations describing a queue is transformed into that of linear equations by the Laplace transform. The resulting system of linear equations is solved by a recursive function called the *potential* (Chydziński, 2007). Chydziński (2006; 2007; 2008) showed how useful this approach is by employing it for calculation of different characteristics such as queue size distribution, loss process, time to buffer overflow, and many others in either the stationary or the transient state.

In our previous work (Rusek *et al.*, 2012) we showed that the approach of Chydziński (2007) can be extended to a system with semi-Markov service times. Derivation of the loss process in $MAP/SM/1/b$ was an example. This paper summarizes previous results and extends them with a new queue characteristic, namely, time to buffer overflow.

The paper is divided into seven sections. Section 2 introduces basic notation and conventions used in the paper. The model of network traffic is presented in Section 3. In Section 4 the queue model is introduced. Section 5 contains the main results of the paper, the proofs of two formulas for the Laplace transform of tail of the time to buffer overflow and the number of losses in a given time period. Computational aspects of the results are also discussed in that section. Section 6 presents some numerical results based on a trace-driven simulation. Finally, Section 7 concludes the paper.

2. Notation

In this paper, both the input process and the service process are related to some Markov process. In order to have an unambiguous notation, each property x of the input process will be denoted by $\check{x}$ while the corresponding property of the service process will be denoted by $\hat{x}$. Note that this convention applies only to properties having counterparts, like the number of states.

In this paper, block matrices appear quite often. Therefore, some shorthand notation will be used. The symbol $\bigoplus$ denotes the *matrix direct sum* (not to be confused with the Kronecker sum)—a shorthand notation for a block diagonal matrix. The Kronecker product will be denoted by $\otimes$.

Transposition of a matrix X will be traditionally denoted by X^T. In order to denote the block transposition in a block matrix, the following property of the transposition operation will be used:

$$\begin{bmatrix} A^T & B^T \\ C^T & D^T \end{bmatrix}^T = \begin{bmatrix} A & C \\ B & D \end{bmatrix}.$$

Column vectors of ones and zeros will be denoted by $\mathbf{1}$ and $\mathbf{0}$, respectively. Traditionally, I is the identity matrix and I_x is the identity matrix of size x.

3. Traffic model

The traffic model (interarrival times and packet lengths) used in this paper can be seen as a special case of the BMAP. In order to make the model analytically tractable for a *packet-oriented* queue, we assumed no correlation between the arrival rate of batches and the batch size.

3.1. Interarrival times. Formally, the BMAP is defined here as a two-dimensional Markov process $(N(t), \check{J}(t))$ on the state space $\{(i,j) : i \geq 0, 1 \leq j \leq \check{m}\}$, with the infinitesimal generator for $(N(t), \check{J}(t))$

equal to

$$Q = \begin{bmatrix} D_0 & D_1 & D_2 & D_3 & \cdots \\ \mathbf{0} & D_0 & D_1 & D_2 & \cdots \\ \mathbf{0} & \mathbf{0} & D_0 & D_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where D_k are $\check{m} \times \check{m}$ matrices, the elements in $D_k, k \geq 1$ are nonnegative, D_0 has nonnegative off-diagonal elements and negative diagonal elements. It is also assumed that $D = \sum_{k=0}^{\infty} D_k$ is an infinitesimal generator and $D \neq D_0$. $N(t)$ denotes the number of arrivals in $(0, t]$ and $\check{J}(t)$ denotes the state of the underlying (modulating) Markov chain with the infinitesimal generator D.

The BMAP can also be defined in the so-called ‘constructive’ way. Assume that the underlying Markov chain has just entered a state i. After an exponentially distributed time with parameter λ_i, a transition to another state occurs. With probability $p_i(0, j)$, there is a transition to state j without a batch arrival. With probability $p_i(k, j)$ there is a transition to state j and simultaneously k jobs arrive. It is assumed that

$$p_i(0, i) = 0, \quad \sum_{k=0}^{\infty} \sum_{j=1}^{\check{m}} p_i(k, j) = 1, \tag{1}$$

and

$$\begin{aligned} \lambda_i &= -(D_0)_{ii}, & & 1 \leq i \leq \check{m}, \\ p_i(0, j) &= \frac{(D_0)_{ij}}{\lambda_i}, & & 1 \leq i, j, i \neq j \leq \check{m}, \\ p_i(k, j) &= \frac{(D_k)_{ij}}{\lambda_i} & & 1 \leq i, j \leq \check{m}. \end{aligned}$$

In relation to IP traffic modeling, batches (simultaneous arrivals of multiple jobs) represent packets of different lengths (in bytes or some other memory units). Thus the memory block implicitly becomes a job in a queue model.

A special case of the BMAP, when $D_k = 0, k > 1$ (no batches), is called the Markovian Arrival Process (MAP). A special case of that MAP, with D_1 being a diagonal matrix, is known as the Markov Modulated Poisson Process (MMPP) (Fischer and Meierhellstern, 1993). The MMPP is also a popular traffic model. However, the MAP is more flexible.

Formulas for many first and second order statistics of the interarrival times in both the MAP and the BMAP can be derived (see the work of Chydziński (2008) for a brief description). Here we introduce only the average arrival rate in the MAP,

$$\lambda = \check{\pi} D_1 \mathbf{1}, \tag{2}$$

where $\check{\pi}$ is the stationary vector for D,

$$\check{\pi} D = \mathbf{0}^T, \quad \check{\pi} \mathbf{1} = 1.$$

This will be important in Section 5, where it is used to obtain the packet loss ratio.

3.2. Packet lengths. The constructive definition of the BMAP is the starting point for a derivation of a packet length model. According to this definition, in each state i of the underlying Markov chain, there is a bivariate transition probability distribution $p_i(k, j)$. This distribution determines packet lengths and the next state of the Markov modulator.

Because the transition distribution is bivariate and each state has a corresponding arrival intensity, it is possible to capture the correlation between packet lengths and the traffic intensity (Klemm *et al.*, 2003). In this paper, we use an approximation and assume that packet lengths and traffic intensity are independent. However, we assume that either of them could be autocorrelated. In such a case, the packet length and the next state are described by the marginal distributions of $p_i(k, j)$. To be more precise, let t_n denote the n-th state transition time of the BMAP. The marginal distributions p_i are given by

$$T_{i,j} = \mathbb{P}(\check{J}(t_{n+1}) = j | \check{J}(t_n) = i) = \sum_k p_i(k, j),$$

and

$$E_{i,k} = \mathbb{P}(l(t_n) = k | \check{J}(t_n) = i) = \sum_j p_i(k, j),$$

where $l \in \{l_1, \ldots, l_{\hat{d}}\}$ is the observable probabilistic outcome in each state. This outcome represents packet lengths and, indirectly, the time required to transfer the packet to the destination. Independent descriptions of packet length and the next state lead to a packet lengths model known as the Hidden Markov Model (HMM) (Rabiner, 1989; Dainotti *et al.*, 2006).

If all state transitions were accompanied by an arrival, such a model could be used as a packet model. However, due to transitions without arrival ($k = 0$), probabilities T_{ij} and E_{ik} have to be modified to remove this unobserved emission. This can easily be done using the total probability theorem in a similar way to that presented in Appendix. However, it is more practical to estimate model parameters directly from the trace file. The resulting HMM model gives an additional flexibility in parameter estimation.

In summary, the proposed HMM packet model is parameterized by the following:

1. $\hat{m}$, the number of states in the model. This parameter corresponds to the number of states in the BMAP.

2. L distinct observation symbols (an alphabet). This parameter corresponds to batch sizes in the BMAP and is the vector of distinct packet lengths.

3. State transition probabilities T, the transition matrix of the underlying Markov chain.

4. Emission probabilities E, the discrete distribution of packet lengths emitted in each state.

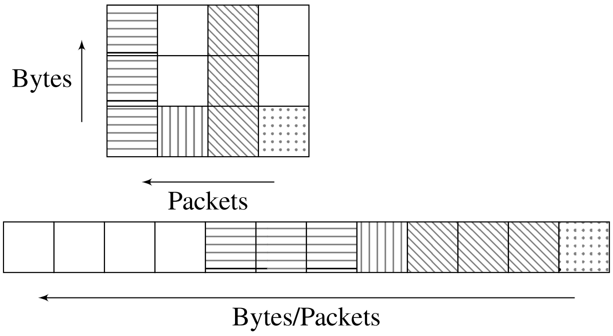


Fig. 1. Difference between packet (top) and byte (bottom) oriented buffers. Service times are represented by the columns. So the packet-oriented buffer requires variable and correlated service times.

4. Queue model

The model of a byte-oriented queue fed by the BMAP is fairly simple. Since a memory block (e.g., 1 B) is a job and the link drains the buffer at a constant number of bytes per unit time, the service time (time to send a single byte or other memory unit) is constant. In this model, the interface is similar to a bucket leaking a liquid. The model can be written as $BMAP/G/1/b$ in Kendall's notation, where b is the total amount of memory.

The model becomes complicated when the buffer is packet-oriented. The queue is still drained at a constant number of bytes per unit time. However, a single byte can no longer be identified as a job. Now, the entire packet is a job and there is no need for a batch arrival (see Fig. 1), therefore the BMAP is reduced to an MAP, i.e., information about packet sizes is removed from the input process. However, this information can be transferred to the service process in an $MAP/SM/1/b$ system, by introducing the states of the server. Each state of the server corresponds to a state of the traffic source in the BMAP. In addition, the states introduce the memory and autocorrelation of packet lengths because each state has an associated service time distribution. This can be visualized in Fig. 1 as a *transformation of batch byte arrival to batch byte removals*.

In this paper, we consider a single server FIFO queuing system fed by the Markovian arrival process, and the service times are assumed to be autocorrelated. The system capacity is finite and equal to b (including a customer currently being served).

Since the service times are simply scaled packet lengths, it is straightforward to derive the service process from the model presented in Section 3.2. The states of the HMM are mapped to those of the server. In the i-th server state, the service time is a random variable with CDF $F_i(t), i = 1, \dots, \hat{m}$. After each service completion, the state of the server is changed in such a way that the state sequence forms a homogenous Markov chain.

Let $\{\hat{J}_n\}, n \geq 0$ denote the state after $n-1$ transitions and $G_n, n \geq 0$ be the $(n-1)$-th service time. If we suppose that $G_0 = 0$, the described service process $(\hat{J}_n, G_n)$ is a semi-Markov process (Janssen and Manca, 2006) defined on a finite set of states $S = \{1, 2 \dots, \hat{m}\}$ by the matrix

$$P_{\hat{J}_{n-1},j}(t) = \mathbb{P}(\hat{J}_n = j, G_n \leq t | (\hat{J}_{n-1}, G_{n-1})).$$

Note that the semi-Markov process is constructed only of service times. The idle periods are ignored. However, we define $\hat{J}(t) \in S$ as the server state at time t including the idle periods. We assume that the time origin corresponds to a departure epoch.

The definition of the semi-Markov process allows dependence of sojourn time on the current state and the future state as well. However, for practical purposes we can assume the sojourn time as independent of the future state. Under this assumption, the HMM of packet lengths can be interpreted as a service process. In such a case $P_{i,j}(t) = T_{i,j} F_i(t)$, where T is the state transition matrix of the underlying Markov chain.[1]

Because of a packet loss, the state transition matrix T of the server is not exactly the same as the transition matrix T in an HMM estimated from traffic. The quantitative analysis of the problem is presented in Appendix. However, for a small loss probability the difference is negligible.

5. Queue characteristics

Chydziński (2006; 2007; 2008) reported numerous queue characteristics of nontrivial queuing systems expressed in terms of 'potential', a recurrent matrix series, defined later in this section. All of his results were derived for a system with uncorrelated service times. This section shows how to generalize the potential method for systems with semi-Markov service times.

Two important characteristics are considered. The first is the time to reach the buffer capacity, also known as the 'first passage time' and abbreviated as *ttbo*, which stands for 'time to buffer overflow'. The second characteristic is the loss process described by the expected number of losses in a given time interval, or the loss ratio. Note that the formula for *ttbo* is also valid for an infinite system. In such a case, first passage time means that the queue exceeded a given threshold.

A key step in the derivation of those characteristics is to modify the counting function. For the MAP, the counting function is defined as follows:

$$\check{P}_{i,j}(n,t) = \mathbb{P}(N(t) = n, \check{J}(t) = j | \\ |N(0) = 0, \check{J}(0) = i). \quad (3)$$

[1] In general, $T = [P_{i,j}(\infty)]_{i,j}$.

Since the state of the service process creates the second degree of freedom in the initial conditions, the counting function has to be redefined in the following way:

$$P_{ijkl}(n,t) = \mathbb{P}(N(t) = n, \check{J}(t) = k, \hat{J}(t) = l| \\ |N(0) = 0, \check{J}(0) = i, \hat{J}(0) = j). \quad (4)$$

It is a fourth rank tensor, and in general it is difficult to compute, unless t is single packet service time counted from the departure epoch. In such a case, assuming independence of the MAP and service processes, the new counting function is simplified to

$$P_{ijkl}(n,t) = T_{jl}\check{P}_{ik}(n,t), \quad (5)$$

where T_{jl} is the server state transition matrix defined in Section 4 and $\check{P}_{ik}(n,t)$ is the standard counting function for the MAP. The product form follows directly from the assumption of the independence of the arrival and service processes.

A modified counting function implicates new forms of some auxiliary matrices:

$$A_{j,k}(s) = \int_0^\infty e^{-st}\check{P}(k,t)\,\mathrm{d}F_j(t), \quad (6)$$

$$\bar{D}_{j,k}(s) = \int_0^\infty e^{-st}\check{P}(k,t)(1 - F_j(t))\,\mathrm{d}t, \quad (7)$$

$$\mathcal{A}_n(s) = \bigoplus_{i=1}^{\hat{m}} A_{i,n}(s) \cdot T \otimes I_{\check{m}}, \quad (8)$$

$$\mathcal{D}_k(s) = [\bar{D}^T_{1,k}(s), \cdots, \bar{D}^T_{\hat{m},k}(s)]^T, \quad (9)$$

$$\bar{\mathcal{A}}_n(s) = \sum_{k=n}^{\infty} \mathcal{A}_k(s), \quad (10)$$

$$Y_k(s) = \left[\frac{\lambda_i p_i(k,j)}{s + \lambda_i}\right]_{ij}, \quad (11)$$

$$\mathcal{Y}_k(s) = \bigoplus_{i=1}^{\hat{m}} Y_k(s), \quad (12)$$

$$z(s) = \Big[\underbrace{\frac{1}{s+\lambda_1}, \dots, \frac{1}{s+\lambda_{\check{m}}}}_{\hat{m} \text{ times}}\Big]^T. \quad (13)$$

In this notation the obtained results are easier to present and more consistent with those for an independent system.

5.1. Time to buffer overflow. Let us start with the queue length at time t denoted as $X(t)$. Let the first time to buffer overflow be denoted by

$$\tau_{n,i,j} = \inf\{t > 0 : X(t) = b| \\ |X(0) = n, \check{J}(0) = i, \hat{J}(0) = j\}. \quad (14)$$

The transform of the tail of $\tau_{n,i,j}$,

$$h_{n,i,j}(s) = \int_0^\infty e^{-st}\mathbb{P}(\tau_{n,i,j} > t)\,\mathrm{d}t. \quad (15)$$

In order to use the standard matrix notation, $h_{n,i,j}(s)$ is turned into a column vector $h_n(s)$ using an inverse lexicographic order:

$$h_n(s) = [h_{n,1,1}(s), \dots, h_{n,\check{m},1}(s), \dots, \\ h_{n,1,\hat{m}}(s), \dots, h_{n,\check{m},\hat{m}}(s)]^T. \quad (16)$$

Theorem 1. *For the $MAP/SM/1/b$ system, if the MAP input process and the SM service process are independent and the matrix T is nonsingular, then*

$$h_n(s) = \sum_{k=0}^{b-n} R_{b-k-n}(s)\mathcal{A}_k(s)h_{b-1}(s) \\ - \sum_{k=1}^{b-n} R_{b-k-n}(s)C_k(s), \quad (17)$$

$$h_{b-1}(s) = W_b^{-1}(s)V_b(s), \quad (18)$$

where the potential $R_k(s)$ is defined as

$$R_0(s) = \mathbf{0}, \quad R_1(s) = \mathcal{A}_0^{-1}(s), \quad (19)$$

$$R_{k+1}(s) = R_1(s)(R_k(s) \\ - \sum_{i=0}^{k} \mathcal{A}_{i+1}(s)R_{k-i}(s)), \qquad k \geq 1 \quad (20)$$

and

$$W_b(s) = (\mathcal{Y}_0(s) - I)\sum_{k=0}^{b} R_{b-k}(s)\mathcal{A}_k(s) \\ + \mathcal{Y}_1(s)\sum_{k=0}^{b-1} R_{b-k-1}(s)\mathcal{A}_k(s), \quad (21)$$

$$V_b(s) = (\mathcal{Y}_0(s) - I)\sum_{k=1}^{b} R_{b-k}(s)C_k(s) \\ + \mathcal{Y}_1(s)\sum_{k=1}^{b-1} R_{b-k-1}(s)C_k(s) - z(s), \quad (22)$$

$$C_n(s) = \sum_{k=0}^{n-1} \mathcal{D}_k(s)\mathbf{1}. \quad (23)$$

Proof. If the system is not empty at the beginning, then from the total probability theorem applied to the first service completion time we get

$$
\begin{aligned}
&\mathbb{P}(\tau_{n,i,j} > t)\\
&\quad = \sum_{\check{k}=1}^{\check{m}} \sum_{\hat{k}=1}^{\hat{m}} \sum_{k=0}^{b-n-1} \int_0^t \mathbb{P}(\tau_{n+k-1,\check{k},\hat{k}} > t-u)\\
&\qquad\qquad \times P_{i,j,\check{k},\hat{k}}(k,u)\,\mathrm{d}F_j(u)\\
&\qquad + (1-F_j(t)) \sum_{\check{k}=1}^{\check{m}} \sum_{k=0}^{b-n-1} \check{P}_{i,\check{k}}(k,t),\\
&\qquad\qquad\qquad\qquad 0 < n \le b, \quad (24)
\end{aligned}
$$

The first term in (24) represents the case when the first service completion time u was before time t and there was no packet loss. In this case, the number of packet arrivals in $(0,u]$ must be less than $b-n-1$; the state of the server might have changed. If the first service completion was before t and there was an overflow, $\mathbb{P}(\tau_{n,i,j} > t) = 0$, so this term does not appear in the equation. The second term corresponds to the situation when the first service completion time u is after time t. This happens with probability $1-F_j(t)$, which depends on the current state of the server.

For an empty system, the total probability formula applied with respect to the first event (arrival or state transition) gives

$$
\begin{aligned}
&\mathbb{P}(\tau_{0,i,j} > t)\\
&\quad = \sum_{\check{k}=1}^{\check{m}} \int_0^t \mathbb{P}(\tau_{0,\check{k},j} > t-u) p_i(0,\check{k}) \lambda_i e^{-\lambda_i u}\,\mathrm{d}u\\
&\quad + \sum_{\check{k}=1}^{\check{m}} \int_0^t \mathbb{P}(\tau_{1,\check{k},j} > t-u) p_i(1,\check{k}) \lambda_i e^{-\lambda_i u}\,\mathrm{d}u\\
&\quad + e^{-\lambda_i t}. \qquad (25)
\end{aligned}
$$

The first and the second terms represent, respectively, a state transition and an arrival. The third component represents no event before time t.

At this point, let us consider only the first term in (24) and see how it behaves under the Laplace transform. It is a convolution, thus in the Laplace transform domain it becomes a product of several terms,

$$
h_{n,i,j}(s) = \sum_{\check{k}=1}^{\check{m}} \sum_{\hat{k}=1}^{\hat{m}} \sum_{k=0}^{b-n-1} h_{n+k-1,\check{k},\hat{k}}(s) T_{j,\hat{k}} [A_{j,k}]_{i,\check{k}},
$$

plus some additional terms omitted for simplicity. If $h_{n+k-1,\check{k},\hat{k}}(s)$ were a matrix, the formula for each column could be rewritten in the form

$$
h_{n,i}(s) = \sum_{\hat{k}=1}^{\hat{m}} \sum_{k=0}^{b-n-1} h_{n+k-1,\hat{k}}(s) T_{j,\hat{k}} A_{j,k} + \cdots .
$$

The above summation is equivalent to the multiplication of a block matrix $\mathcal{A}_k$ and the flattened $h_{n+k-1,\check{k},\hat{k}}(s)$. Similar block matrix composition can be applied to other terms in (24) and (25), resulting in the following form after the Laplace transform:

$$
\begin{aligned}
h_n(s) = \sum_{k=0}^{b-n-1} \mathcal{A}_k(s) h_{n+k-1}(s) \qquad (26)\\
+ C_{b-n}(s), \quad 0 < n \le b, \qquad (27)
\end{aligned}
$$

$$
h_0(s) = \mathcal{Y}_0(s) h_0(s) + \mathcal{Y}_1(s) h_1(s) + z(s). \qquad (28)
$$

From this point we can follow the proof of Theorem 3.2.1 by Chydziński (2007) simply by replacing $h_n(s) = u_{b-n}(s)$. With these new variables, Eqns. (26) and (28) can be rewritten in the form

$$
\begin{aligned}
&\sum_{k=-1}^{n-1} \mathcal{A}_{k+1}(s) u_{n-k}(s)\\
&\qquad - u_n(s) = \psi_n(s), \quad 0 < n \le b, \qquad (29)
\end{aligned}
$$

$$
u_b(s) = \mathcal{Y}_0(s) u_b(s) + \mathcal{Y}_1(s) u_{b-1}(s) + z(s), \qquad (30)
$$

where $\psi_n(s) = \mathcal{A}_n u_1(s) - C_n(s)$. Lemma 3.2.1 by Chydziński (2007) gives the closed-form solution

$$
u_n(s) = R_n(s) c(s) + \sum_{k=1}^{n} R_{n-k}(s) \psi_k(s), \qquad (31)
$$

where $c(s)$ is a vector independent of n. This vector can be calculated by putting $n = 1$ into (31), which gives $c(s) = \mathcal{A}_0 u_1(s)$, and the solution for u_n gets the form

$$
u_n(s) = R_n(s) \mathcal{A}_0 u_1(s) + \sum_{k=1}^{n} R_{n-k}(s) \psi_k(s). \qquad (32)
$$

By means of the boundary condition (28), it is straightforward to derive Eqns. (17) and (18).

The last thing to be proven is why $\mathcal{A}_0$ is nonsingular, a condition required by Lemma 3.2.1 of Chydziński (2007). Using properties of the Kronecker product matrix product and the matrix direct sum, it is straightforward to find that

$$
\det \mathcal{A}_0 = (\det T)^{\check{m}} \prod_{i}^{\hat{m}} \det A_{i,0}. \qquad (33)
$$

It is discussed by Chydziński (2007) why $A_{i,0}$ are nonsingular. Thus the matrix T is the only source of singularities and this completes the proof of Theorem 1. ■

5.2. Loss process. Let $L(t)$ denote the number of losses observed in an interval $(0,t]$, and $\Delta_{n,i,j}(t)$ be its average value conditioned on the initial queue length and the initial states of the MAP and the SM service process, namely,

$$\begin{aligned}\Delta_{n,i,j}(t) &= \mathbb{E}(L(t)|X(0)=n, \check{J}(0)=i, \hat{J}(0)=j),\\ &0 \le n \le b, \quad 1 \le i \le \check{m} \quad 1 \le j \le \hat{m}. \qquad (34)\end{aligned}$$

The Laplace transform of $\Delta_{n,i,j}(t)$ is denoted by $\delta_{n,i,j}(s)$, namely,

$$\delta_{n,i,j}(s) = \int_0^\infty e^{-st}\Delta_{n,i,j}(t)\,\mathrm{d}t. \qquad (35)$$

In order to use the standard matrix notation, $\delta_{n,i,j}(s)$ is turned into a column vector similar to (16).

Theorem 2. *If the MAP input process and the SM service process are independent and the matrix T is nonsingular, then the Laplace transform of the average number of losses in $(0,t]$ in the $MAP/SM/1/b$ queue has the form*

$$\begin{aligned}\delta_n(s) = &\Big(R_{b-n+1}(s)\mathcal{A}_0(s)\sum_{k=0}^{b-n}R_{b-n-k}(s)B_k(s)\Big)\\ &\times M_b^{-1}(s)l_b(s)\\ &+\sum_{k=0}^{b-n}R_{b-n-k}(s)g_k(b,s), \qquad (36)\end{aligned}$$

where

$$\begin{aligned}B_n(s) &= \mathcal{A}_{n+1}(s) - \bar{\mathcal{A}}_{n+1}(s)\bar{\mathcal{A}}_0^{-1}(s),\\ g_k(s) &= \bar{\mathcal{A}}_{k+1}(s)\bar{\mathcal{A}}_0^{-1}(s)c_b(s) - c_{b-k}(s),\\ c_k(s) &= \frac{1}{s}\sum_{i=b-k}^{\infty}(i-b+k)\mathcal{A}_i(s)\cdot\mathbf{1}\\ &\quad + \sum_{i=b-k}^{\infty}(i-b+k)\mathcal{D}_i(s)\cdot\mathbf{1}, \qquad (37)\\ l_b(s) &= \mathcal{Y}_1(s)\sum_{k=0}^{b-1}R_{b-k-1}(s)g_k(s)\\ &\quad - (I-\mathcal{Y}_0(s))\sum_{k=0}^{b}R_{b-k}(s)g_k(s), \qquad (38)\\ M_b(s) &= (I-\mathcal{Y}_0(s))\\ &\quad\times\left(\sum_{k=0}^{b}R_{b-k}(s)B_k(s) + R_{b+1}(s)\mathcal{A}_0(s)\right)\\ &\quad - \mathcal{Y}_1(s)\Big(\sum_{k=0}^{b-1}R_{b-k-1}(s)B_k(s)\\ &\quad + R_b(s)\mathcal{A}_0(s)\Big). \qquad (39)\end{aligned}$$

Proof. If the system is not empty at the beginning, then from the total probability theorem applied to the first service completion time we get

$$\begin{aligned}&\Delta_{n,i,j}(t)\\ &\quad= \sum_{\check{k}=1}^{\check{m}}\sum_{\hat{k}=1}^{\hat{m}}\sum_{k=0}^{b-n-1}\int_0^t\Delta_{n+k-1,\check{k},\hat{k}}(t-u)\\ &\quad\times P_{i,j,\check{k},\hat{k}}(k,u)\,\mathrm{d}F_j(u)\\ &\quad+\sum_{\check{k}=1}^{\check{m}}\sum_{\hat{k}=1}^{\hat{m}}\sum_{k=b-n}^{\infty}\int_0^t(k-b+n \qquad (40)\\ &\quad+\Delta_{b-1,\check{k},\hat{k}}(t-u))P_{i,j,\check{k},\hat{k}}(k,u)\,\mathrm{d}F_j(u)\\ &\quad+(1-F_j(t))\sum_{\check{k}=1}^{\check{m}}\sum_{k=b-n}^{\infty}(k-b+n)\\ &\quad\times\check{P}_{i,\check{k}}(k,t), \quad 0<n\le b.\end{aligned}$$

The first term in (40) represents the case when the first service completion time u was before time t and there was no packet loss. In this case, the number of packets arrivals in $(0,u]$ must be less than $b-n-1$; the state of the server might have changed.

The second term in (40) represents the case when the first service completion time u was before time t and there was a buffer overflow. This means that over the time interval $(0,u]$ we had $k \ge b-n$ arrivals and $k-b+n$ of them were dropped; the state of the server might have changed.

Finally, the third term in (40) corresponds to the situation when the first service completion time u is after time t. This happens with probability $1-F_i(t)$, which depends on the current state of the server. In this case, the mean number of losses is given by

$$\sum_{\check{k}=1}^{\check{m}}\sum_{k=b-n}^{\infty}(k-b+n)\check{P}_{i,\check{k}}(k,t).$$

Let us now suppose that at time $t=0$ the queue is empty. Applying the total probability formula with respect to the first arrival time, we obtain

$$\begin{aligned}&\Delta_{0,i,j}(t)\\ &\quad=\sum_{k=1}^{\check{m}}\int_0^t\Delta_{0,k,j}(t-u)p_i(0,k)\lambda_i e^{-\lambda_i u}\,\mathrm{d}u \qquad (41)\\ &\quad+\sum_{k=1}^{\check{m}}\int_0^t\Delta_{1,k,j}(t-u)p_i(1,k)\lambda_i e^{-\lambda_i u}\,\mathrm{d}u.\end{aligned}$$

The first term in (41) corresponds to the case where the first MAP state transition is not accompanied by an arrival, while the second term corresponds to the simultaneous transition and an arrival.

Applying the Laplace transform and matrix notation to Eqns. (40) and (41), we get

$$\begin{aligned}\delta_n(s) = &\sum_{k=0}^{b-n-1} \mathcal{A}_k \delta_{n+k-1}(s)\\ &+ \sum_{k=b-n}^{\infty} \mathcal{A}_k \delta_{b-1}(s) + c_n(s), \quad 0 < n \le b,\end{aligned} \tag{42}$$

$$\delta_0(s) = \mathcal{Y}_0(s)\delta_0(s) + \mathcal{Y}_1(s)\delta_1(s). \tag{43}$$

From this point the proof is *mutatis mutandis* the same as that of Theorem 1. ■

5.3. Transient and stationary analysis. Theorems 1 and 2 give the Laplace transforms of queuing characteristics. However, following Chydziński (2007) in exploiting the limiting properties of the Laplace transform, we can compute the stationary parameters as well.

Let us begin with the first passage time. Theorem 1 can be used to calculate the expected value of $\tau_{n,i,j}$ as well as the transform of its distribution function,

$$\mathbb{E}(\tau_{n,i,j}) = h_{n,i,j}(0), \tag{44}$$

$$\int_0^\infty e^{-st}\mathbb{P}(\tau_{n,i,j} < t)\,\mathrm{d}t = \frac{1}{s} - h_{n,i,j}(s). \tag{45}$$

Numerical values of the distribution can be computed by applying an algorithm (e.g., Abate *et al.*, 2000) for the Laplace transform inversion. The expected number of losses at some particular time $\Delta_{n,i,j}(t)$ can be computed in the same way. The stationary packet loss ratio (LR) formula is given by

$$LR = \lim_{t\to\infty} \frac{1}{\lambda}\frac{\mathrm{d}\Delta_{n,i,j}(t)}{\mathrm{d}t} = \lim_{s\to 0+} \frac{s^2\delta_{n,i,j}(s)}{\lambda}. \tag{46}$$

Because the stationary loss ratio does not depend on the initial condition, we can choose any n, i and j. For some other characteristics, such as local loss intensity, see the results of Chydziński (2008).

5.4. Computational aspects. The system of equations (42) and (43) can be solved in at least two different ways. The first one is the potential method, and the second one is a direct approach. In the direct approach, (42) and (43) are combined into a single matrix equation,

$$\mathfrak{A}\cdot\delta(s) = \mathfrak{B}, \tag{47}$$

where

$$\delta(s) = [\delta_0^T(s), \dots, \delta_b^T(s)]^T, \tag{48}$$

$$\mathfrak{A} = \begin{bmatrix} \mathcal{Y}_0 & \mathcal{Y}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0}\\ \mathcal{A}_0 & \cdots & \cdots & \mathcal{A}_{b-2} & \bar{\mathcal{A}}_{b-1} & \mathbf{0}\\ \mathbf{0} & \mathcal{A}_0 & \dots & \mathcal{A}_{b-3} & \vdots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \ddots & \mathcal{A}_0 & \bar{\mathcal{A}}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \bar{\mathcal{A}}_0 & \mathbf{0} \end{bmatrix} - I, \tag{49}$$

$$\mathfrak{B} = [\mathbf{0}, -c_1^T, \cdots, -c_b^T]^T. \tag{50}$$

In (49) $\mathbf{0}$ denotes the zero square matrix of size $\check{m}\times\hat{m}$ while in (50) it is the vector of the same size. The linear system (49) can be solved even for a singular T.

The main advantage of the potential method over the direct approach is its lower numerical complexity equal to $\check{m}^3\hat{m}^3b^2$ as compared to $\check{m}^3\hat{m}^3b^3$ for the direct approach. Despite its lower complexity, the potential method may sometimes be harder in realisation, because it often requires high precision computations (higher than the machine precision).

A practical application of Theorems 1 and 2 requires an effective algorithm for computing matrices $A_k(s)$ and $\bar{D}(s)$.

Direct computation of the MAP counting function is problematic. Therefore, the uniformization technique has been developed for efficient numerical calculations (Lucantoni, 1991). Using uniformization, Eqns. (6) and (7) for a particular state can be rewritten to following forms:

$$A_k(s) = \sum_{j=0}^\infty \gamma_j(s)K_{k,j}, \tag{51}$$

$$\bar{D}_k(s) = \sum_{j=0}^\infty \beta_j(s)K_{k,j}, \tag{52}$$

where

$$\gamma_j(s) = \int_0^\infty e^{-(s+\theta)t}\frac{(\theta t)^j}{j!}\,\mathrm{d}F(t), \tag{53}$$

$$\beta_j(s) = \int_0^\infty e^{-(s+\theta)t}\frac{(\theta t)^j}{j!}(1-F(t))\,\mathrm{d}t, \tag{54}$$

$$\theta = \max_i\,(-D_0)_{ii}.$$

For the simplicity of notation, the first indices of $A_{j,k}(s)$ and $\bar{D}_{j,k}(s)$ were omitted, since it is not important for which service state the matrices are computed. The matrices $K_{k,j}$ can be easily computed using the recursive formulas presented by Lucantoni (1991). The main advantage of using uniformization, besides its computational aspects, is that if a service time distribution changes, only (53) and (54) need to be computed.

The important case is when the service time is a discrete random variable. This is the case for a real traffic model where packet lengths are discrete. Before applying uniformization to such a distribution, let us consider a more general case, i.e., the mixture of distribution.

Let the service time distribution be a mixture of some distributions $F^{(i)}(t)$ with weights p_i, i.e., $F(t) = \sum_i p_i F^{(i)}(t)$. For each i, let $\gamma_j^{(i)}(s)$ and $\beta_j^{(i)}(s)$ denote respectively the functionals (53) and (54) for the i-th component of the mixture.

Proposition 1. *For the mixture distribution $F(t)$, the functionals (53) and (54) get the form*

$$\gamma_j(s) = \sum_i p_i \gamma_j^{(i)}(s), \tag{55}$$

$$\beta_j(s) = \sum_i p_i \beta_j^{(i)}(s). \tag{56}$$

The discrete service time is nothing more than a mixture of some deterministic service times. Therefore, all we need to know is the form of the functionals (53) and (54) for the deterministic service time d. For this case the integrals (53) and (54) can be calculated analytically and are equal to

$$\gamma_j = \frac{e^{-(\theta+s)d}(\theta d)^j}{j!}, \tag{57}$$

$$\beta_j = \theta^j \frac{\Gamma(j+1,0) - \Gamma(j+1, d(s+\theta))}{j!(s+\theta)^{j+1}}, \tag{58}$$

where $\Gamma(j,x)$ is the incomplete gamma function. Now, using Proposition 1 combined with Eqns. (57) and (58), it is straightforward to apply uniformization and compute matrices $A_k(s)$ and $\bar{D}_k(s)$.

6. Numerical results

In this section, we present the queue characteristics obtained for parameters estimated from the actual network traffic generated by the dormitories of the AGH University of Science and Technology (Cracow, Poland). We captured 150,000 IP packets from the uplink. The traffic was used to fit the model parameters and to run a trace-driven simulation. The simulation results were later compared with the model predictions. Buffer parameters of the router were taken from experiments reported by Rusek *et al.* (2011). The queue was simulated in our own simulator written in MATLAB and Java, the same as used by Rusek *et al.* (2011). The correctness of the simulator was verified by comparison with ns-2.

6.1. Parameter estimation. There are many fitting algorithms designed for Markovian traffic models (cf., e.g., Klemm *et al.*, 2003; Heyman and Lucantoni, 2003; Muscariello *et al.*, 2005). We found the Expectation Maximisation (EM) algorithm (Klemm *et al.*, 2003) to fit our needs. The model estimated by the EM algorithm can accurately approximate first and second order statistics of the traffic while being quite simple (small dimensionality) and numerically tractable. Keeping model simplicity and accuracy under consideration, we modelled the traffic as a four state ($\check{m} = 4$) MAP.

The HMM of packet lengths was parameterized in a similar way by means of a Baum–Welch algorithm implemented in the MATLAB function *hmmtrain* (Rabiner, 1989). However, the packet lengths require some preprocessing because the real traffic contains quite a large number (hundreds) of different packet lengths. It is obvious that such a fine-grained trace cannot be handled by any numerical procedure.

The problem has to be changed to have just a few distinct packet lengths. This can be done quite easily since the major parts of a packet are either small ($\approx$ 50 B) or large ($\approx$ 1500 B), with a small percentage forming the medium group ($\approx$ 600 B). In the work of Klemm *et al.* (2003) the entire length spectrum was divided into a small number of ranges, and for each range the mean length was used. Here, we take a different approach. We run a k-means clustering algorithm on packet lengths to find the course-grained representation. The algorithm finds the best centroid points L as well as group boundaries. The only parameter is the number of centroids. Having seen that most packets are either small, medium or large, three centroids is the recommended choice.

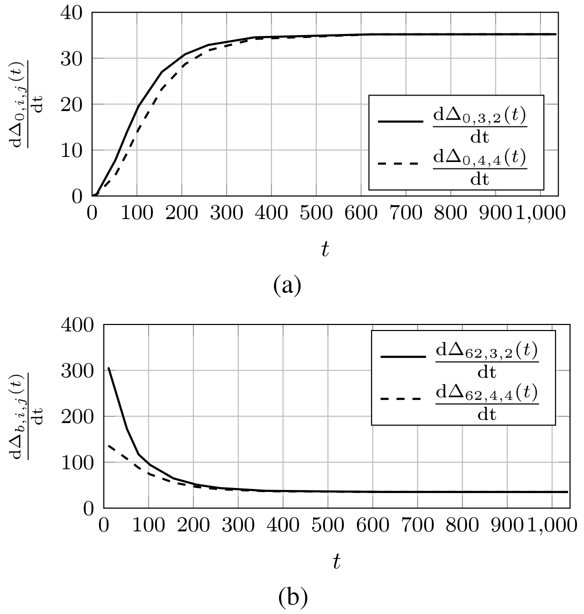
Packet lengths after clusterization are now suitable for fitting an HMM. Since the Baum–Welch algorithm may not converge to a global minimum, it is often necessary to run the algorithm multiple times to find the best parameters. The key is to match the autocorrelation function of packet lengths as closely as possible, sometimes at the cost of a higher error of mean or variance.

Because an n-state BMAP can describe both the packet lengths and the interarrivals, we may expect that the MAP and the HMM should have a similar ($\sim n$) number of states. Indeed, experimenting with different numbers of states and emissions, we found that a four state HMM with two emissions ($\hat{d} = 2$) was enough to fit the autocorrelation of packet lengths. A two-state HMM could also match the autocorrelation function, but not as often as the four state model. The model parameterization in all numerical examples used hereafter is as follows:

$$D_0 = \begin{bmatrix} -11300. & 35.9 & 20.6 & 1.28 \times 10^{-6} \\ 0.297 & -2290. & 34.6 & 12.3 \\ 0.00147 & 2.55 & -503. & 1.38 \\ 216. & 2.99 & 243. & -8590. \end{bmatrix},$$

$$D_1 = \begin{bmatrix} 10100. & 559. & 524. & 0.0202 \\ 70.2 & 1850. & 233. & 88.8 \\ 0.0378 & 22.6 & 412. & 64.3 \\ 130. & 237. & 3790. & 3970. \end{bmatrix},$$

$$T = \begin{bmatrix} 0.840 & 0.0754 & 0.0352 & 0.0491 \\ 0.222 & 0.757 & 0.0123 & 0.00836 \\ 0.0386 & 0.0151 & 0.573 & 0.374 \\ 0.00249 & 0.0384 & 0.573 & 0.386 \end{bmatrix},$$



(a)

(b)

Fig. 2. Local intensity of the loss process for the initially empty (a) and full (b) buffer ($b = 62$ packets). The time unit is the average service time.

$$E = \begin{bmatrix} 0.868 & 0.132 \\ 0.201 & 0.799 \\ 0.994 & 0.00617 \\ 0.985 & 0.0145 \end{bmatrix}, \quad L = \begin{bmatrix} 82 & 1307 \end{bmatrix}.$$

Note that D_0 and D_1 contains only rates and none of them is a infinitesimal generator. However, $D = D_0 + D_1$ is a generator, and the sum of its rows equals zero. The independent model uses an empirical discrete distribution. However, the stationary distribution of the HMM can be used instead.

6.2. Model validation. The main advantage of the proposed model is the ability to compute transient state characteristics of the system. One example of such a characteristic is the loss intensity shown in Fig. 2. The plots are made for two of sixteen different pairs of the initial states and two extreme buffer sizes (empty and full) for a one-second-long time interval. The transient state analysis gives a unique chance to analyze the maximal loss intensities as well as the time to steady state. In the presented example the steady state begins after about 300 service times on the average, and after that time the loss intensity is constant and equal about 35 drops per second. Note that during the transient state the loss intensity could be more than ten times greater.

It is possible to compute the queue characteristics on a typical PC for quite large buffers, but an important question is how accurate the model is and how it behaves with real-sized buffers. Measuring the queuing characteristics of a real packet buffer may be difficult or even impossible without deep interaction with the hardware. However, the study of buffering in the Cisco ISR router interface described by Rusek *et al.* (2011) allows the building of an accurate simulator of a real buffer.

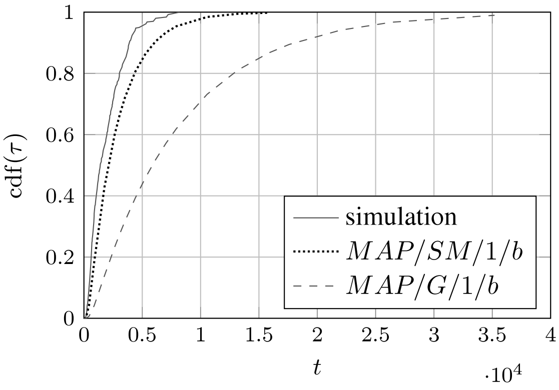


Fig. 3. Cumulative distribution function of time to reach the buffer of size 168 starting an empty system. The utilization $\rho = 0.95$, states of the MAP and the server are according to the stationary distribution. The time unit is the average service time.

Stationary loss process characteristics (e.g., the loss ratio) are easy to measure by counting each lost packet. With *ttbo* this is more difficult because the buffer has to be prepared in a given state, e.g., $n = 0$. In our approach we randomly selected 500 points in the trace. Then the queue was simulated starting at the selected point. When the system was empty, a timer started, and it was stopped when the buffer was full. The average value of all observations was used as the estimator of *ttbo*, and confidence intervals were obtained by the bootstrap method.

Figure 3 shows the prediction of two models (with independent and correlated service times) compared with simulation results for a heavily loaded queue. In each case the system is initially empty and the states of the MAP and the server are according to the stationary distribution.

The $MAP/SM/1/b$ model is not perfect, but it is much more accurate than a model with independent service times. The accuracy increases for smaller buffers (see Fig. 4) for both models. This is to be expected, as it takes only few service times to fill the small buffer, so the service time correlation becomes less important.

A large queue with autocorrelated service can be filled much faster compared to a system with an independent one. This is easily observed in Fig. 5, showing the average value of the time to buffer overflow, and it becomes even more clear as the load drops in Fig. 6. In this case the *ttbo* value is about 3 s and the model predicts 13 s, while the old model with independent services predicts the buffer will be full after 4 min.

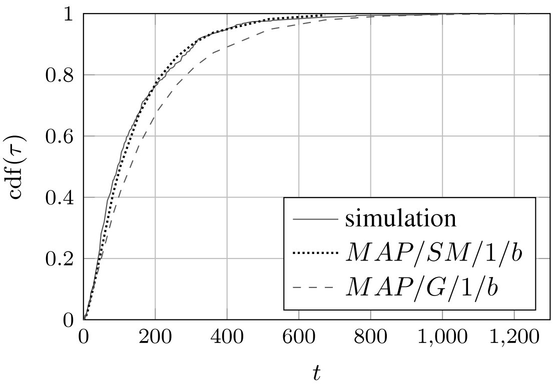


Fig. 4. Cumulative distribution function of time to reach the buffer of size 20 starting an empty system. The utilization $\rho = 0.8$, states of the MAP and the server are according to the stationary distribution. The time unit is the average service time.

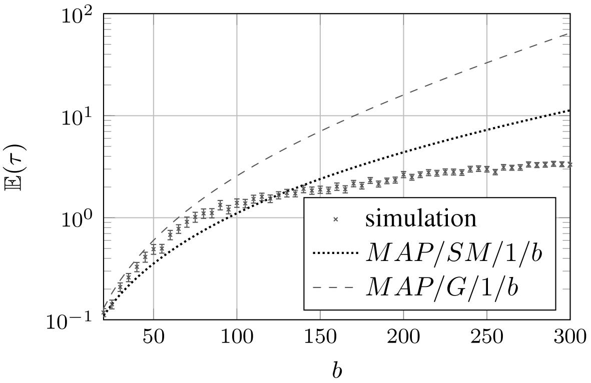


Fig. 5. Average time to buffer overflow versus the buffer size. The utilization $\rho = 0.95$, states of the MAP and the server are according to the stationary distribution.

Again, for small buffers both models give similar results and the real value is somewhere inbetween. For large buffers, on the other hand, the models give extremely different predictions (up to two orders of magnitude). Exactly the same situation is observed for the stationary packet loss ratio in Fig. 7. This clearly indicates that the autocorrelation function of service times has a huge impact on the queuing characteristics, especially for large buffers. Therefore, the autocorrelation in service times is vital as some modern devices have buffers larger than 128 packets (Rusek *et al.*, 2011).

The $MAP/SM/1/b$ model is superior to $MAP/G/1/b$ in all aspects but is still inaccurate for very large buffers. This is probably caused by neglecting long range dependence. The estimated MAP was observed to be accurate up to some time scale (33 ms), and above this scale it is no longer accurate. Similar behaviour was observed for the HMM and packet lengths. This should explain the model's inaccuracy for large buffers, because the larger the buffer the further back in time it 'looks', thus the long range autocorrelation becomes important.

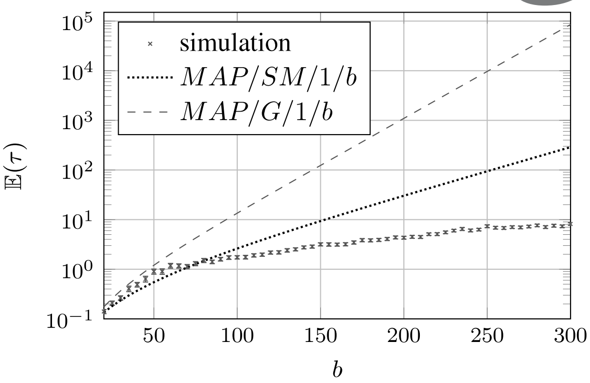


Fig. 6. Average time to buffer overflow versus the buffer size. The utilization $\rho = 0.8$, states of the MAP and the server are according to the stationary distribution.

The fact that packet lengths and interarrival times are not independent as the model assumes may also be important. However, this is the least probable explanation since the Pearson correlation coefficient (Weisstein, 2013) was only about -0.1.

Although a better estimation technique would increase the accuracy, we cannot model a true long range dependence by a Markov model. It will always be just an approximation but, having said that, sometimes an approximate result is enough. A perfect match to the trace file gives nothing because the TCP changes traffic in response to each packet loss. Approximate predictions, on the other hand, can still be informative.

7. Conclusion

Correlation of service times in queuing systems has a great impact on the characteristics of such systems and cannot be omitted. This is especially important in the case of packet-oriented network devices, because packet lengths are highly autocorrelated. Modelling packet lengths as the HMM allows the construction of a solvable mathematical model of such a device.

The proposed model is a queuing system with a Markovian arrival process and semi-Markov service times. Queuing characteristics obtained for such a system are quite accurate approximations of the characteristics evaluated for a real device. Moreover, such a model is much more accurate compared with previous approaches assuming independent service times. The main contribution of this paper is the theoretical analysis of this system.

The results include closed-form representations of transforms of the loss process and the tail distribution of first passage time. They were derived using a novel

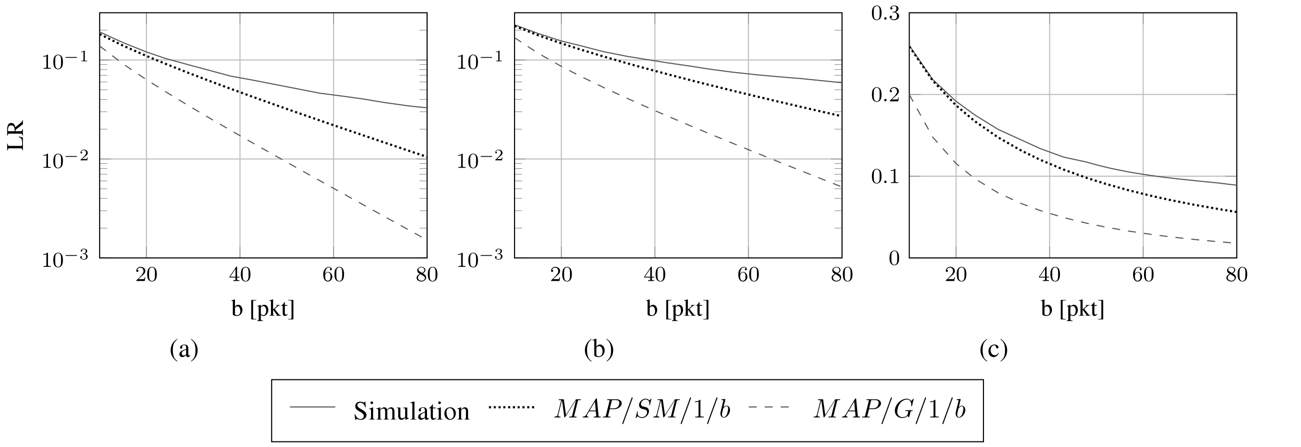


Fig. 7. Stationary packet loss ratio versus the buffer size for $\rho = 0.7$ (a), $\rho = 0.8$ (b) and $\rho = 0.9$ (c). Confidence intervals ($\approx 2\%$) are omitted because they are tiny and invisible on the logarithmic scale.

approach to queuing systems with autocorrelated service times. Derived formulae are suitable either for the transient or the stationary analysis and allow computation of the following queue characteristics: expected time to reach the buffer capacity and its distribution, the loss ratio, local loss intensity or the total number of losses. It is possible to extend the obtained results beyond the loss process and compute e.g., queue size distribution, in a similar way as in the work of Chydziński (2006).

Acknowledgment

This research was supported in part by the PL-Grid Infrastructure and the national project 647/N-COST/2010/0.

References

Abate, J., Choudhury, G. and Whitt, W. (2000). An introduction to numerical transform inversion and its application to probability models, *in* W. Grassmann (Ed.), *Computational Probability*, International Series in Operations Research & Management Science, Vol. 24, Springer, New York, NY, pp. 257–323.

Chydziński, A. (2006). Queue size in a BMAP queue with finite buffer, *in* Y. Koucheryavy, J. Harju and V. Iversen (Eds.), *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Lecture Notes in Computer Science, Vol. 4003, Springer, Berlin/Heidelberg, pp. 200–210.

Chydziński, A. (2007). Time to reach buffer capacity in a BMAP queue, *Stochastic Models* **23**(2): 195–209.

Chydziński, A. (2008). Packet loss process in a queue with Markovian arrivals, *7th International Conference on Networking, ICN 2008, Cancum, Mexico*, pp. 524–529.

Chydziński, A. and Chróst, Ł. (2011). Analysis of AQM queues with queue size based packet dropping, *International Journal of Applied Mathematics and Computer Science* **21**(3): 567–577, DOI: 10.2478/v10006-011-0045-7.

Dainotti, A., Pescape, A., Salvo Rossi, P., Iannello, G., Palmieri, F. and Ventre, G. (2006). Qrp07-2: An HMM approach to internet traffic modeling, *IEEE Global Telecommunications Conference, GLOBECOM'06, San Francisco, CA, USA*, pp. 1–6.

Dudin, A.N., Klimenok, V.I. and Tsarenkov, G.V. (2002). A single-server queueing system with batch Markov arrivals, semi-Markov service, and finite buffer: Its characteristics, *Automation and Remote Control* **63**(8): 1285–1297.

Dudin, A.N., Shaban, A.A. and Klimenok, V.I. (2005). Analysis of a queue in the BMAP/G/1/N system, *International Journal of Simulation Systems, Science & Technology* **6**(1–2): 13–22.

Emmert, B., Binzenhöfer, A., Schlosser, D. and Weiß, M. (2007). Source traffic characterization for thin client based office applications, *in* A. Pras and M. van Sinderen (Eds.), *Dependable and Adaptable Networks and Services*, Lecture Notes in Computer Science, Vol. 4606, Springer, Berlin/Heidelberg, pp. 86–94.

Fischer, W. and Meierhellstern, K. (1993). The Markov-modulated Poisson process (MMPP) cookbook, *Performance Evaluation* **18**(2): 149–171.

Heyman, D. and Lucantoni, D. (2003). Modeling multiple IP traffic streams with rate limits, *IEEE/ACM Transactions on Networking* **11**(6): 948–958.

Janowski, L. and Owezarski, P. (2010). Assessing the accuracy of using aggregated traffic traces in network engineering, *Telecommunication Systems* **43**(3–4): 223–236.

Janssen, J. and Manca, R. (2006). *Applied Semi-Markov Processes*, Springer, Science+Business Media, New York, NY.

Klemm, A., Klemm, E., Lindemann, C. and Lohmann, M. (2003). Modeling IP traffic using the batch Markovian arrival process, *Performance Evaluation* **54**(2): 149–173.

Kobayashi, H. and Ren, Q. (1992). A mathematical-theory for transient analysis of communication networks, *IEICE Transactions on Communications* **E75B**(12): 1266–1276.

Lucantoni, D.M. (1991). New results on the single server queue with a batch Markovian arrival process, *Communications in Statistics: Stochastic Models* **7**(1): 1–46.

Lucantoni, D.M., Choudhury, G.L. and Whitt, W. (1994). The transient BMAP/G/L queue, *Communications in Statistics. Stochastic Models* **10**(1): 145–182.

Meyer, C. (2000). *Matrix Analysis and Applied Linear Algebra*, Miscellaneous Titles in Applied Mathematics Series, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Muscariello, L., Mellia, M., Meo, M., Marsan, M.A. and Cigno, R.L. (2005). Markov models of internet traffic and a new hierarchical MMPP model, *Computer Communications* **28**(16): 1835–1851.

Neuts, M.F. (1966). The single server queue with poisson input and semi-Markov service times, *Journal of Applied Probability* **3**(1): 202–230.

Paxson, V. and Floyd, S. (1995). Wide area traffic: The failure of Poisson modeling, *IEEE/ACM Transactions on Networking* **3**(3): 226 –244.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE* **77**(2): 257 –286.

Robert, S. and LeBoudec, J. (1996). On a Markov modulated chain exhibiting self-similarities over finite timescale, *Performance Evaluation* **27**(8): 159–173.

Rusek, K., Janowski, L. and Papir, Z. (2011). Correct router interface modeling, *Proceedings of the 2nd Joint WOSP/SIPEW International Conference on Performance Engineering, ICPE'11, Karlsruhe, Germany*, pp. 97–102.

Rusek, K., Papir, Z. and Janowski, L. (2012). MAP/SM/1/b model of a store and forward router interface, *2012 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Genova, Italy*, pp. 1–8.

Salvador, P., Pacheco, A. and Valadas, R. (2004). Modeling IP traffic: Joint characterization of packet arrivals and packet sizes using BMAPs, *Computer Networks* **44**(3): 335–352.

Schwefel, H.-P., Lipsky, L. and Jobman, M. (2001). On the necessity of transient performance analysis in telecommunication networks, *Teletraffic Engineering in the Internet Era: Proceedings of the International Teletraffic Congress, ITC-17, Salvador da Bahia, Brazil*, pp. 1087–1099.

Sequeira, L., Fernandez-Navajas, J., Saldana, J. and Casadesus, L. (2012). Empirically characterizing the buffer behaviour of real devices, *2012 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Genova, Italy*, pp. 1–6.

Weisstein, E.W. (2013). Correlation coefficient, From MathWorld—A Wolfram web resource, `http://mathworld.wolfram.com/CorrelationCoefficient.html`.

Krzysztof Rusek is a Ph.D. candidate at the Department of Telecommunications, AGH University of Science and Technology. He received an M.Sc. in electronics and telecommunications in 2009 from the same university. His main interest is performance evaluation of telecommunications systems and queuing theory. He is also interested in computer vision and machine learning. He has actively participated in several 7th FPs and national scientific projects (COST Action IC0703, INDECT, INSIGMA, TAPAS).

Lucjan Janowski is an assistant professor at the Department of Telecommunications, AGH University of Science and Technology. He received his M.Sc. in 2002 and his Ph.D. in telecommunications in 2006, both from the AGH University of Science and Technology. In 2007 he worked at a post-doc position in CNRS-LAAS (Laboratory for Analysis and Architecture of Systems of CNRS) in France, where he prepared both malicious traffic analysis and anomaly detection algorithms. In 2010/2011 he spent half a year at a post-doc position at the University of Geneva, working on QoE for health applications. His main interests are statistics as well as probabilistic modelling of subjects and subjective rates used in QoE evaluation. He has participated in several commercial and scientific projects.

Zdzisław Papir is a professor at the Department of Telecommunications, AGH University of Science and Technology in Cracow, Poland. He received the M.Sc. degree in electrical engineering in 1976 and the Ph.D. degree in computer networks, both from the AGH University of Science and Technology. In 1992 he received the postdoctoral degree from the Technical University of Gdańsk. In the years 1991–98 he made several visits at universities in Belgium, Germany, Italy, and US, working on statistical modelling of telecommunication traffic. In the years 1994–95 he served for the Polish Cable Television as the Network Design Department manager. He has (co)authored five books and about 100 research papers. Between 1999–2006 he was a guest editor for *IEEE Communications Magazine* responsible for the Broadband Access Series. He has participated in several R&D IST European projects (COST, Eureka Celtic, and the 4th, 5th, 6th and 7th FPs (ACTS, ESPRIT, IST, ICT), being personally responsible for statistical modelling of telecommunication traffic and quality assessment of multimedia communication services. He has also been appointed as an expert in information and communications technologies by the European Commission.

Appendix

In the HMM of packet lengths there is a state transition upon each packet arrival. If the packet is dropped, the information about transition is lost. Therefore the real service process is slightly different from the HMM estimated from the traffic. Note that only the transition matrix T is changed.

Let T be the transition matrix estimated from the traffic and T' be the transition matrix of the service process. If drops can be assumed to be independent, then it is possible to estimate the parameters of the real service times by virtue of the following proposition.

Proposition A1. *For each packet, if drop probabilities are independent and equal to p, then the service time transition matrix is given by*

$$T' = (1-p)T + p(1-p)T^2(I-pT)^{-1}. \tag{A1}$$

Proof. Applying the total probability theorem to the number of successive drops, we get

$$\begin{aligned} T' &= (1-p)T + \underbrace{p(1-p)T^2 + p^2(1-p)T^3 + \cdots}_{\text{unobserved transitions}} \qquad \text{(A2)} \\ &= (1-p)T + p(1-p)T^2(I + pT + p^2T^2 + \cdots). \end{aligned}$$

The first part of (A2) represents the situation when the state transition was observed. This happens with the probability $1-p$. In this case the transition matrix is unchanged.

The second part of (A2) represents the situations when there were $1, 2, 3, \ldots$ drops (unobserved successive state transitions). This happens with the probability $(1-p)p$, $(1-p)p^2$, $(1-p)p^3, \ldots$. In those cases the service time transition matrix corresponds to the $2, 3, 4, \ldots$-step transition matrix of the estimated process.

The matrix T is stochastic and $p < 1$, so each eigenvalue of pT is less than 1. Therefore, the second part can be written in a closed form using the Neumann series formula (Meyer, 2000) which gives (A1). ■

The real loss ratio can be found as a fixed point of the following map: $p = LR(T'(p))$. The best approach is to use the estimated T as an approximation and compute LR. In the next steps we use the previously computed LR to find the new T using (A1) and improve accuracy.

To show some numerical results of this procedure, we simulated the queue with an artificially generated MAP and HMM. The observed loss ratio was $LR_{sim} = 0.2176$ and its 95% confidence interval was $(0.2166, 0.2185)$. The first approximation was $LR_1 = 0.2243$. The next iterations gave $LR_2 = 0.2171$, $LR_3 = 0.2173$ and $LR_4 = 0.2173$. The procedure converges after about four iterations and the result is mostly within the confidence interval.

The assumption about drop independence is strong and not always satisfied. Therefore there is no guarantee nor proof that this procedure improves accuracy. It might even not converge at all. However, we did not observe such a case. In practice the improvement is negligible compared with some other errors, for example, measurements errors (Rusek *et al.*, 2011), and it is fine to use the estimated T in calculations.

Received: 10 May 2013
Revised: 7 October 2013