Analysis of the Discrete-time Multi-queue System with a Cycle-based Scheduler

Wojciech Burakowski and Maciej Sosnowski

National Institute of Telecommunications, Warsaw, Poland

Abstract—This paper presents an analysis of a discrete-time multi-queue system handling a number of packet streams. The analysis focuses on calculating system state distribution and packet sojourn time distribution. The method relied upon for determining system state distribution is based on creating a number of equations that are solved numerically. Next, based on the distribution calculated in such a manner, we derive relations for packet sojourn time distribution. The models studied may be useful for instance in a system supporting a number of virtual links (each of a constant bitrate) that share a common physical link. Isolation of performance of those virtual links needs to be assured. Finally, we present some exemplary numerical results showing the usefulness of the proposed analysis for supporting the system dimensioning process.

Keywords—*discrete-time queueing system with vacations, system state distribution, packet sojourn time distribution, virtualized system.*

1. Introduction

The paper presents an analysis of a FIFO-type discrete-time queueing system handling a number of packet streams in which service access of specific streams is governed by a cycle that has been assumed *a priori*. The cycle is repeated periodically and consists of a number of time slots, dedicated to handling packets assigned to a predefined stream. Therefore, from the point of view of a given stream, it is the system with vacations in which the so-called active state and vacation periods may be distinguished.

Packets belonging to a given stream may only be serviced in the active periods (time spent serving packets from this stream), while in the vacation periods, these packets cannot be serviced. The important feature here is that such an approach guarantees performance isolation between the packet streams handled. It means that performance-related parameters, i.e. delays and loss of packets belonging to a given stream, are not disturbed by servicing other streams. The idea behind a system with vacations fed by one packet stream is illustrated in Fig. 1. The system may operate in one of two potential states, i.e.: active period A, when

packet service is available, and vacation period V, when packet service is not available.
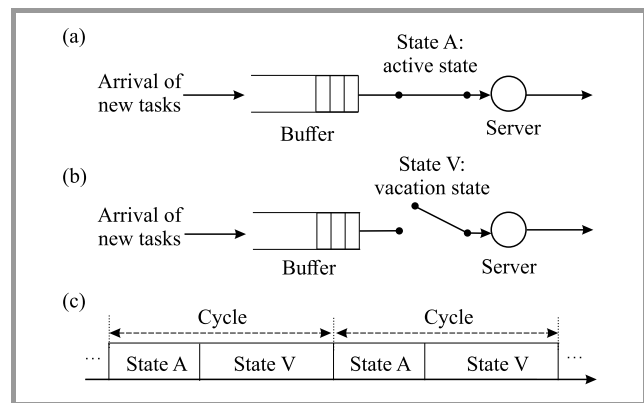
Fig. 1. Block diagram of a system with vacations fed by one packet stream: (a) in state A (active period), (b) in state V (vacation period), (c) states A and V alternate accordingly to a cycle assumed on an *a priori* basis.

The system under analysis is a model of a solution supporting a number of virtual links that share one physical link. Hence, performance isolation between packet streams within the specific virtual links is required.

Figure 2 shows an example of two virtual links established between virtualized servers, each with two virtual machines (VM). One virtual link is dedicated to transferring packets from VM1 to VM3, while the other is dedicated to transferring packets from VM2 to VM4. Such an approach to a virtualized system was successfully implemented and tested, for instance, in the IIP System [1] that was designed for creating a number of parallel Internets (with different protocol stacks) sharing the same physical resources, i.e. physical links and virtualization-enabling devices.

The system analyzed in this paper is of the FIFO discrete-time type, with a constant time of servicing packets from a given stream. These packets are serviced only in specific time slots within the active periods, as new packets from the considered stream may arrive into the system at the beginning of each slot only. For the vacation periods, we assume that they consist of a number of time slots (named
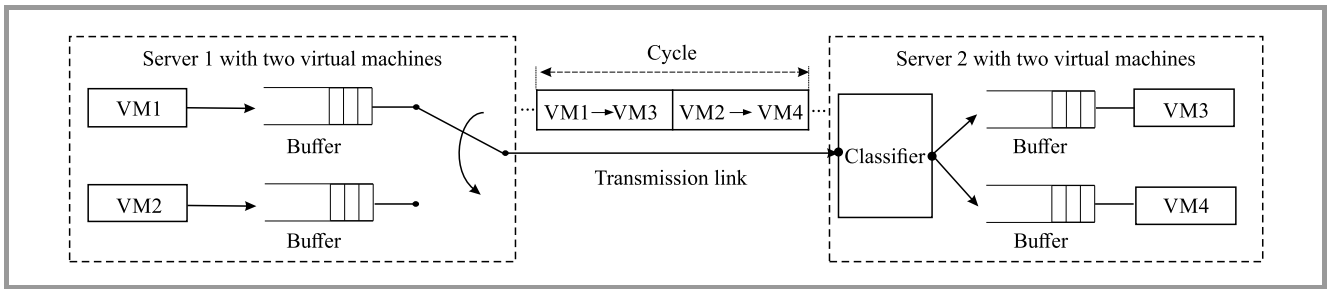
Fig. 2. Scenario with two virtual links established between two virtualized machines: virtual link 1 for data transfer between VM1 and VM3, and virtual link 2 for data transfer between VM2 and VM4.
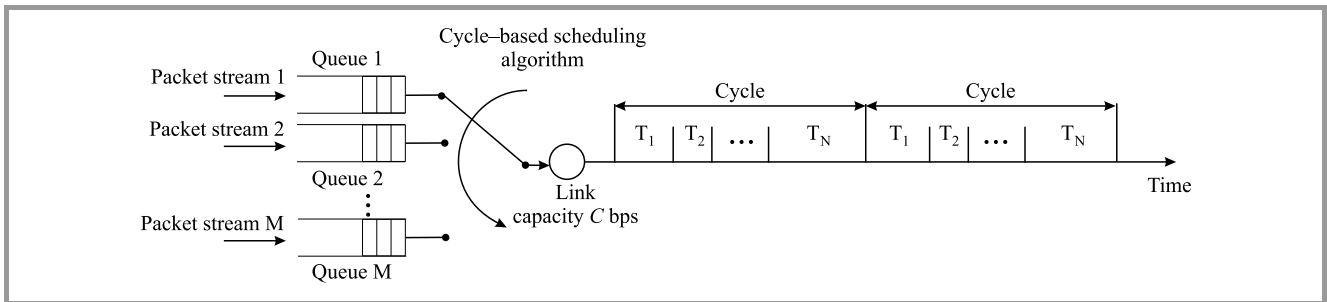

Fig. 3. The proposed discrete-time multi-queue system fed by M independent packet streams.

dummy slots) of the same length as the active periods. Again, new packets arriving during the vacation periods may arrive only at the beginning of dummy slots as well. For such a system with an infinite buffer size, we propose methods allowing to calculate state distribution by numerically solving a number of equations. Then, calculation of sojourn time distribution is proposed as well, using analytical formulas being an extension of the method described in [2], concerning a system without vacations.

Systems with vacations have been studied by many authors making different assumptions. A decent survey of these methods may be found e.g. in [3]–[8]. Unexpectedly, according to our best knowledge, no analysis is available of the system that is discussed in this paper.

Let us mention the papers that directly correspond to the research problem discussed in this paper. In [9], the authors consider a system with vacations but with continuous time and present formulas for mean waiting times. In [10], the authors extend the analysis for the system with general service times of packets and derive relations for system state distribution and packet loss ratio (with a finite buffer). An approximated method for calculating mean waiting times in the considered system fed by a Poissonian stream was proposed in [11]. Finally, an extension of the cycle-based scheduler, capable of providing service with a lower priority for tasks during periods that were not dedicated to them was described in [12].

2. Details of the Studied System

The system under consideration belongs to the family of discrete-time queueing systems with vacations that are fed by M independent packet streams (as shown in Fig. 3), with their buffer being of the infinite size. Access that packets belonging to specific streams have to a commonly shared link of capacity C is governed by a cycle-based scheduler. For this purpose, the system allocates a buffer to each packet stream and assigns a period of time in consecutive cycles. More precisely, cycle time duration T is divided into M periods of T_m $(m = 1, \ldots, M)$, where $T = \sum_{m=1}^{M} T_m$. During the T_m period, only packets belonging to stream m may be transmitted.

The system studied may be analyzed from the point of view of each separate packet stream. This is possible thanks to the use of the cycle-based scheduler which ensures performance isolation between the packet streams serviced. The above means that packet transfer characteristics (defined, for instance, by delay and loss rate) concerning a given packet stream are not disturbed by servicing packets belonging to the remaining streams.

Figure 4 shows the discussed system from the point of view of packet stream m $(m = 1, \ldots, M)$. This packet stream identifies its packet queue and those periods during the consecutive cycles in which the packets from this stream may be served. So, stream m identifies its own cycle in which active period T_{mA} and vacation period T_{mV} may be observed. It needs to be noted that the cycle visible to each packet stream is of the same length as the length of the cycle in the scheduler. The further analysis assumes that the duration of the cycle, as well as the duration of the periods dedicated to serving packets from particular streams, are constant.

Moreover, we assume that packets belonging to the same stream have a constant length. For the sake of simplicity,
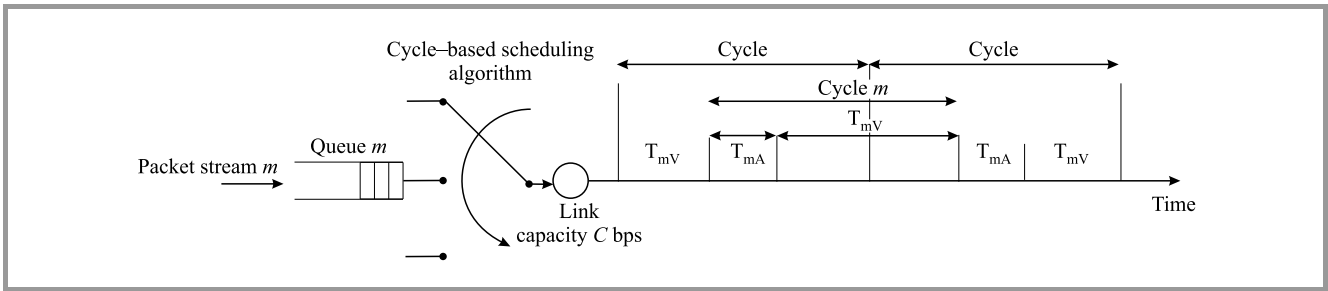
Fig. 4. System from the point of view of handling packets belonging to stream *m*.
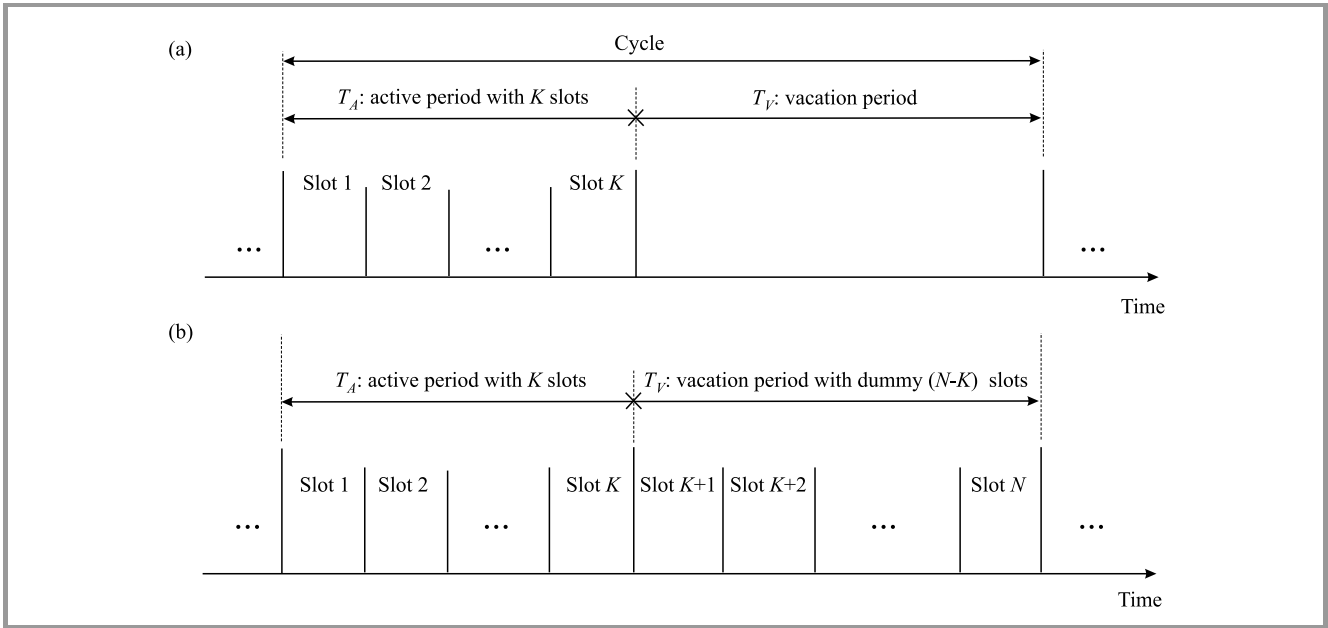

Fig. 5. Time slots within the cycle: (a) real system, *K* time slots in the active period, (b) model proposed for the analysis, with a cycle consisting of *N* slots, *K* slots in the active period, and *N–K* dummy slots in the vacation period.
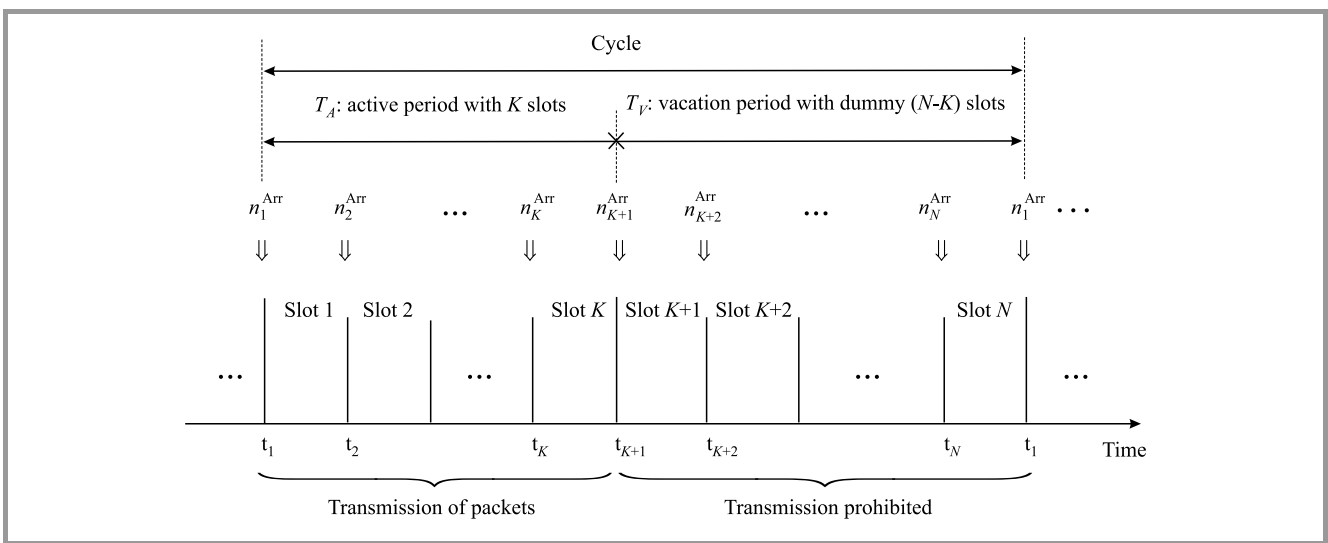

Fig. 6. Packet arrival and service processes.

the numbering of packet streams is omitted here, since the system under consideration is the same from the point of view of each stream. Thus, packets belonging to a given stream are transmitted within a time slot belonging to active periods only, as shown in Fig. 5a. On the other hand, new packets may arrive into the system any time. In order to unify the considerations, we introduce dummy time slots into the vacation periods. The lengths of these time slots are the same as the lengths of time slots in the active periods, as shown in Fig. 5b. However, no packet transmission is allowed in the dummy time slots. Such an assumption is made to count packet sojourn times in a number of slots. Finally, the system under consideration may be classified as a discrete-time system with vacations, where the cycle consists of K time slots in an active period and N time slots in total.

New packets arrive into the system in batches, at the beginning of each slots t_i $(i = 1, \ldots, N)$ only. These moments occur just after the previous slot has finished, and just before the first waiting packet is taken for transmission (for slots from 1 to K only). Therefore, at these moments, all packets in the system wait in a queue (for service). The arrival process of new packets and the service process is illustrated in Fig. 6, where n_i^{Arr} is the number of new packets arriving into the system at time t_i. We analyze the system assuming that the packet arrival process at specific t_i moments may differ for different i, $i = (1, \ldots, N)$.

3. Analysis

In our analysis, we will consider a system with an infinite buffer size. The first observation is that the system maintains constant properties with respect to time slots that have the same position within the cycle. The following relation takes place:

$$\Pr\{k \text{ packets in the system at time } t_i\} =$$
$$\Pr\{k \text{ packets in the system at time } t_i + \tau\},$$
$$\text{where } \tau = j(T_A + T_V), \; j = 1, 2, \ldots. \quad (1)$$

3.1. System State Distribution

The system state is described by the number of packets available in the system at a given time. Let us define the system state n_i $(n_i = 0, 1, 2, \ldots,)$ when the i-th time slot $(i = 1, \ldots, N)$ in the cycle starts. As only one packet may be served during one time slot belonging to the active period, and due to the fact that no packet service occurs during the vacation period, we can write the following relations for those periods (see Fig. 6):

$$\begin{cases} n_i = \max\left(n_{(i-1)} - 1, 0\right) + n_i^{Arr}, & \text{for } i = 2, \ldots, K+1 \\ n_i = n_{(i-1)} + n_i^{Arr}, & \text{for } i = K+2, \ldots, N, 1 \end{cases},$$
$$(2)$$

where n_i^{Arr} denotes the number of new packets arriving into the system at t_i $(i = 1, \ldots, N)$ and $n_{(i-1)}$ is the number of

packets being in the system at the time slot that is located before the i-th time slot (e.g. time slot N is before the time slot 1).

For the sake of simplicity, let us continue our analysis under the assumption that the number of packets arriving into the system at moments t_i $(i = 1, \ldots, N)$ is described by the same probability distribution function. Therefore, probabilities $\Pr\{n_i^{Arr} = j\}$, $j = 0, 1, 2, \ldots$, do not depend on the position of the slot in the cycle. The above assumption is not critical in our approach. In fact, the presented method can be easily adapted to a scenario in which packet arrival processes are not the same for different t_i. However, for a given i these distributions need to be identical. Thanks to Eq. (2), we can write the following set of equations for $i = 2, \ldots, K+1$:

$$\left[\Pr\{n_i = 0\} \; \Pr\{n_i = 1\} \; \Pr\{n_i = 2\} \; \ldots\right] =$$
$$\left[\Pr\{n_{(i-1)} = 0\} \; \Pr\{n_{(i-1)} = 1\} \; \Pr\{n_{(i-1)} = 2 \ldots\}\right] \cdot A, \quad (3)$$
$$\text{for } i = 2, \ldots, K+1,$$

where:

$$A = \begin{bmatrix} \Pr\{n_i = 0 | n_{(i-1)} = 0\} & \Pr\{n_i = 1 | n_{(i-1)} = 0\} & \ldots \\ \Pr\{n_i = 0 | n_{(i-1)} = 1\} & \Pr\{n_i = 1 | n_{(i-1)} = 1\} & \ldots \\ \Pr\{n_i = 0 | n_{(i-1)} = 2\} & \Pr\{n_i = 1 | n_{(i-1)} = 2\} & \ldots \\ \ldots & \ldots & \ldots \end{bmatrix}.$$

Assuming that the arrival process does not depend on the system state, we count the items of matrix A as:

$$\Pr\{n_i = m | n_{(i-1)} = k\} =$$
$$\begin{cases} \Pr\{n^{Arr} = m\}, & \text{for } n = 0 \\ \Pr\{n^{Arr} = m - k + 1\}, & \text{for } k > 0 \text{ and } m \geq k-1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Taking into account Eqs. (3) and (4), matrix A is:

$$A = \begin{bmatrix} \Pr\{n^{Arr} = 0\} & \Pr\{n^{Arr} = 1\} & \Pr\{n^{Arr} = 2\} & \ldots \\ \Pr\{n^{Arr} = 0\} & \Pr\{n^{Arr} = 1\} & \Pr\{n^{Arr} = 2\} & \ldots \\ 0 & \Pr\{n^{Arr} = 0\} & \Pr\{n^{Arr} = 1\} & \ldots \\ 0 & 0 & \Pr\{n^{Arr} = 0\} & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{bmatrix}.$$

Similarly to Eq. (3), using Eq. (2), we can write the following equations for $i = K+2 \ldots, N, 1$:

$$\left[\Pr\{n_i = 0\} \; \Pr\{n_i = 1\} \; \Pr\{n_i = 2\} \; \ldots\right] =$$
$$\left[\Pr\{n_{(i-1)} = 0\} \; \Pr\{n_{(i-1)} = 1\} \; \Pr\{n_{(i-1)} = 2 \ldots\}\right] \cdot B,$$
$$\text{for } i = K+2, \ldots, N, 1, \quad (5)$$

where:

$$B = \begin{bmatrix} \Pr\{n_i = 0 | n_{(i-1)} = 0\} & \Pr\{n_i = 1 | n_{(i-1)} = 0\} & \ldots \\ \Pr\{n_i = 0 | n_{(i-1)} = 1\} & \Pr\{n_i = 1 | n_{(i-1)} = 1\} & \ldots \\ \Pr\{n_i = 0 | n_{(i-1)} = 2\} & \Pr\{n_i = 1 | n_{(i-1)} = 2\} & \ldots \\ \ldots & \ldots & \ldots \end{bmatrix}.$$

Still, assuming that the arrival process does not depend on the system state, we count the items of matrix B in the following manner:

$$\Pr\{n_i = m | n_{(i-1)} = k\} =$$
$$\begin{cases} \Pr\{n^{Arr} = m - k\}, & \text{for } m \geq k \\ 0, & \text{otherwise} \end{cases}. \quad (6)$$

Matrix B is:

$$B = \begin{bmatrix} \Pr\{n^{Arr} = 0\} & \Pr\{n^{Arr} = 1\} & \Pr\{n^{Arr} = 2\} & \cdots \\ 0 & \Pr\{n^{Arr} = 0\} & \Pr\{n^{Arr} = 1\} & \cdots \\ 0 & 0 & \Pr\{n^{Arr} = 0\} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}.$$

By combining Eqs. (3) and (5), we can write the formulas for the distribution of the number of packets in the system at time t_1:

$$\begin{bmatrix} \Pr\{n_1 = 0\} \, \Pr\{n_1 = 1\} \, \Pr\{n_1 = 2\} \, \ldots \end{bmatrix} =$$
$$\begin{bmatrix} \Pr\{n_1 = 0\} \, \Pr\{n_1 = 1\} \, \Pr\{n_1 = 2\} \, \ldots \end{bmatrix} \cdot A^K \cdot B^{(N-K)}. \quad (7)$$

Finally, on the basis of Eq. (7) together with:

$$\sum_{k=0}^{\infty} \Pr\{n_N = k\} = 1, \quad (8)$$

we get the number of equations that can be used to numerically calculate system state distribution at time t_1 – see Eq. (2). On the basis of these values, we can calculate system state distributions for the remaining times t_i $(i = 2, \ldots, N)$, using Eqs. (3) and (5).

Matrices A and B are of an infinite size, due to the unlimited buffer size. However, in practice, we can limit the size of these matrices assuming that we consider the probabilities of packet number arrivals that are greater than the assumed threshold ε, e.g. $\varepsilon \geq 0.0001$. The formula describing the probability that k packets are present in the system is:

$$\Pr\{n = k\} = \frac{1}{N} \sum_{i=1}^{N} \Pr\{n_i = k\}, \quad (9)$$

where $\Pr\{n_i = k\}$ is the probability that in the i-th slot $(i = 1, \ldots, N)$ k packets are present in the system.

3.2. Packet Sojourn Time Distribution

The packet sojourn time is defined as the period between the arrival of a packet into the system and the completion of its service. Here, it is counted for a number of time slots. Packet sojourn time distribution is computed as a function of system state distribution and packet arrival distribution. The analysis begins by recalling the formula that was derived for the packet sojourn time in the case of the discrete-time queueing system without vacations and FIFO discipline fed by packets arriving to the system in each slot accordingly to the same probability distribution [2]:

$$\Pr\{D = k\} = \begin{cases} 0, & \text{for } n = 0 \\ \dfrac{\Pr\{X = k\}}{\rho}, & \text{for } k > 0 \end{cases}, \quad (10)$$

where: $\Pr\{D = k\}$ denotes the probability that packet sojourn time is equal to k $(k = 1, 2, \ldots)$ time slots and $\Pr\{X = k\}$ denotes the probability that at a time immediately following the end of a slot (and just before a packet is taken into service in the next slot), there are k packets in the system, and ρ is the load of each slot.

Unfortunately, Eq. (10) cannot be adapted to the presented system comprising both active and vacation periods, as the load varies between slots and depends also on the slot's position in the cycle. In particular, when packets arrive into the system in each slot based on the same probability distribution, then the first active slot after a vacation period has a greater load than other active slots remaining in the cycle. Equation (10) is nevertheless useful for checking the correctness of a more general formula in which we assume that no vacation periods are present in the system under consideration.

Our approach focuses on those packets that finish their services within given time slots that are a part of the active period. The sojourn time of such packets is equal to the number of time slots. We need to know the time slot during which this packet arrived into the system and how many packets had to be served earlier, i.e. between the moment it arrived into the system and the moment it is taken for service. When the FIFO queuing approach is relied upon, those packets that arrive into the system after the arrival of the packet in question exert no impact on its sojourn time. Thus, a strict dependency exists between the number of packets waiting in the queue at the arrival of the new packet into the system and its sojourn time. Notice that at the period between the arrival moment of a packet and the moment of ending its service the system is in the busy period, meaning that packets are served in all time slots within the active periods in the interval under consideration.

Let us now focus on a packet that is taken into the service in the i-th time slot $(i = 1, 2, \ldots, K)$ and its sojourn time in the system is k time slots. For such a packet, we define two parameters (Fig. 7): position $d_n(i)$ of the time slot in the cycle at the time of arrival of the packet and the number of time slots in the active periods $d_{n_A}(i)$ located between the packet's arrival and the beginning of its service. Notice that the value $d_{n_A}(i)$ denotes the number of waiting packets being present in the system at the arrival of the packet in question. These packets are served before the packet under consideration, due to the fact that the FIFO order of precedence has been adopted. The values of parameters $d_n(i)$ and $d_{n_A}(i)$ are calculated in the following manner:

$$d_n(i) = N - \text{mod}[k - i - 1, N], \text{ for } i = 1, \ldots, K, \quad (11)$$

where $\text{mod}[x, y]$ is the modulo function, and:

$$d_{n_A}(i) = \begin{cases} k, & \text{for } i \geq k \\ i + \left\lfloor \dfrac{k-i}{N} \right\rfloor K + \\ \max(\text{mod}(k-i, N) - (N-K), 0), & \text{for } i < k \end{cases}, \quad (12)$$

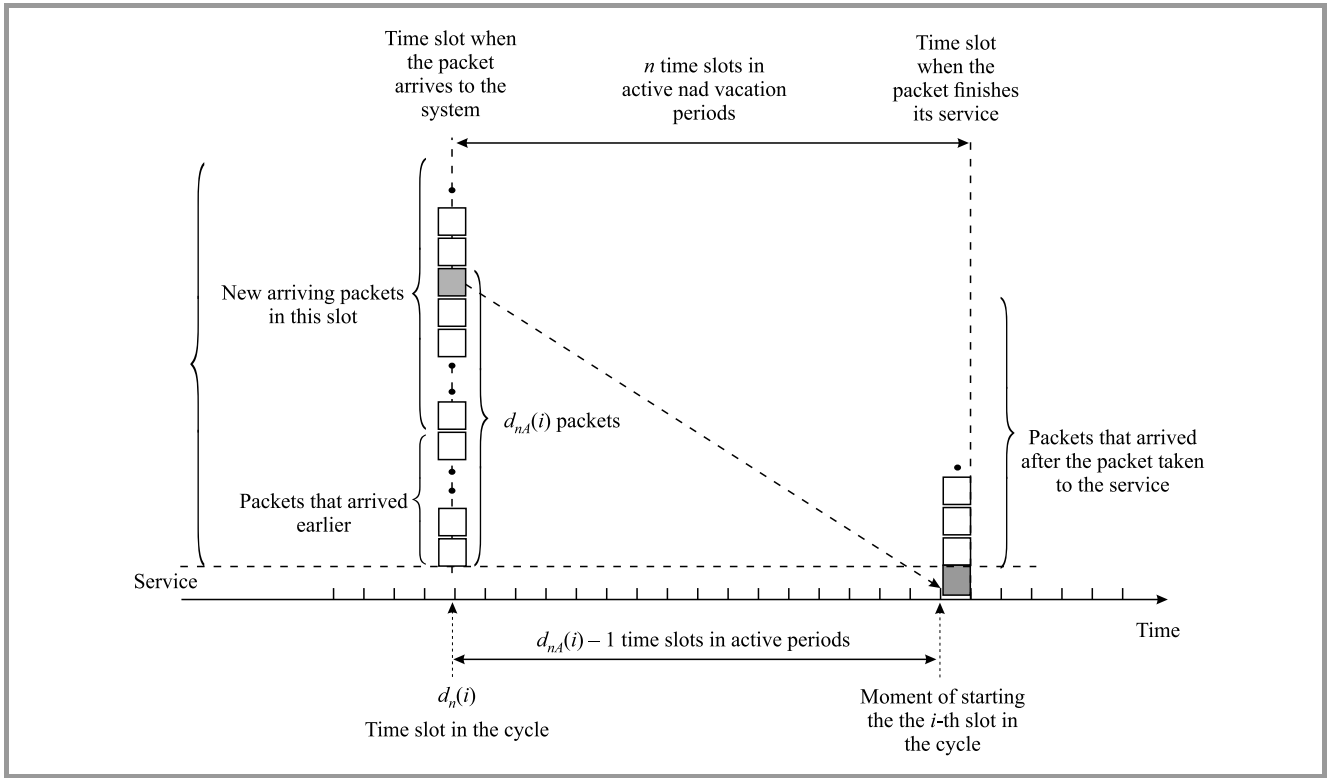where $\lfloor x \rfloor$ is an integer part of x.

Fig. 7. Illustration of parameters $d_n(i)$ and $d_{n_A}(i)$.

To calculate the probability that the sojourn time of the packet which has finished its service in the i-th time slot $(i = 1, \ldots, K)$ and has spent k time slots in the system we use the formula:

For $k = 0$

$$\Pr\{D_i = k\} = 0 \qquad . \quad (13)$$

For $k \geq 1$ and $d_n(i) = 2, \ldots, K+1$

$$\Pr\{D_i = k\} =$$

$$\frac{\left(\begin{array}{l} [\Pr\{X_{d_n(i)-1}=0\}+\Pr\{X_{d_n(i)-1}=1\}]\Pr\{n_i^{Arr} \geq d_{n_A}(i)\}+ \\ \displaystyle\sum_{m=1}^{d_n(i)-1} \Pr\{X_{d_n(i)-1}=m+1\}\Pr\{n_i^{Arr} \geq d_{n_A}(i)-m\} \end{array} \right)}{\rho_i} . \quad (14)$$

For $k \geq 1$ and $d_n(i) = K+2, \ldots, N, 1$

$$\Pr\{D_i = k\} =$$

$$\frac{\left(\begin{array}{l} \Pr\{X_{d_n(i)-1}=0\}\Pr\{n_i^{Arr} \geq d_{n_A}(i)\}+ \\ \displaystyle\sum_{m=1}^{d_n(i)-1} \left[\Pr\{X_{d_n(i)-1}=m\}\Pr\{n_i^{Arr} \geq d_{n_A}(i)-m\} \right] \end{array} \right)}{\rho_i} , \quad (15)$$

where:

- $d_n(i)$ is calculated by Eq. (11),

- $d_{n_A}(i)$ comes from Eq. (12),

- $\Pr\{X_i = m\}$ $(m = 1, \ldots, N)$ denotes the probability that the number of packets in the i-th time slot $(i = 1, \ldots, N)$ equals m,

- $\Pr\{n_i^{Arr} \geq l\}$ $(l = 0, 1, \ldots; \ i = 1, \ldots, N)$ denotes the probability that in the i-th time slot at least l new packets arrive,

- ρ_i is the load of the i-th time slot $(i = 1, \ldots, K)$ such as:

$$\rho_i = 1 - \Pr\{X_i = 0\} . \quad (16)$$

The index $d_n(i) - 1$ from Eqs. (13)–(15) refers to the time slot that precedes the time slot indexed by $d_n(i)$:

$$d_n(i) - 1 := \begin{cases} N, & \text{if } d_n(i) = 1 \\ d_n(i) - 1, & \text{otherwise} . \end{cases} \quad (17)$$

Finally, the probability that packet sojourn time in the system lasts k time slots is:

$$\Pr\{D = k\} = \begin{cases} 0, & \text{for } k = 0, \\ \displaystyle\sum_{i=1}^{K} \frac{\rho_i}{\sum_{j=1}^{K} \rho_j Pr} \{D_i = k\}, & \text{for } k > 0 . \end{cases} \quad (18)$$

Equations (13)–(15) should transform to Eq. (10) when there are no vacation periods in the analyzed system. In this case, we do not distinguish the positions of particular time slots. Moreover, the following relation takes place:

$$d_n = d_{n_A} = k . \quad (19)$$

From Eqs. (13)–(15) we get:

$$\Pr\{D=k\} =$$

$$\begin{cases} 0, & \text{for } k=0, \\ \dfrac{\left(\begin{array}{l}[\Pr\{X=0\}+\Pr\{X=1\}]\Pr\{n^{Arr}\geq k\}+ \\ \displaystyle\sum_{m=1}^{k-1}\Pr\{X=m+1\}\Pr\{n^{Arr}\geq k-m\}\end{array}\right)}{\rho}, & \text{for } k>0 \end{cases} \quad (20)$$

For this system, we can write the following system state distribution equations:

$$\Pr\{X=k\} = (\Pr\{X=0\}+\Pr\{X=1\})\Pr\{n^{Arr}=k\}+$$
$$\sum_{m=1}^{k}\Big[\Pr\{X=m+1\}\Pr\{n^{Arr}=k-m\}\Big],$$
$$\text{for } k=0,1,2,\dots \quad (21)$$

By applying Eq. (21) to:

$$\Pr\{n^{Arr}\geq k\} = \Pr\{n^{Arr}\geq k-1\} - \Pr\{n^{Arr}=k-1\},$$
$$\text{for } k=1,2,\dots \quad (22)$$

the identity of Eqs. (20) and (10) for the system without vacations can be proved (see Appendix A).

3.3. Numerical Examples

Here, results corresponding to the system are presented with 15 time slots per cycle, with the first 5 slots constituting the active period ($T_A = 5$) and the remaining 10 slots belonging to the vacation period ($T_V = 10$). We will compare system state distributions and packet sojourn time distributions under the assumption that the utilization factor of the active periods equals 0.9 and that the packet arrival distribution is the same in each slot. Two following scenarios are verified:

- 1 – the packets arrive into the system according to binomial distribution with $p = 0.3$, i.e. $\Pr\{n^{Arr}=1\} = 0.3$ and $\Pr\{n^{Arr}=0\} = 0.7$.
- 2 – the packets arrive into the system according to geometric distribution with $p = 0.23$, $\Pr\{n^{Arr}=k\} = (1-p)p^k$ for $k = 0,1,2,\dots$.

For scenario 2, the variance of the arrival process distribution is greater than in scenario 1. The characteristics describing system state distributions for the two scenarios under consideration are presented in Figs. 8 and 9.

Figure 8 shows the mean $E[n_i]$ and the variance $var[n_i]$ of the random variable holding the number of packets being in the system at t_i $(i = 1, 2, \dots, N)$. These values are greater in scenario 2 than in scenario 1. As expected, the lowest mean values of the number of packets in the system are observed at the beginning of vacation periods (slot 6), while their maximum values occur at the beginning of active periods (slot 1).

Figures 9 and 10 show the system state distribution calculated from Eq. (9) and the packet sojourn time distribution calculated from Eq. (18), respectively. These distributions may be characterized by long tails, since some packets

may be kept waiting for their service even for a number of cycles.

The numerical results were confirmed by a discrete event simulation software written by the authors. The simulator validates both system state distribution at t_i $(i = 1, \dots, N)$ and packet sojourn time distribution. Results of the simulations are not presented, as they are indistinguishable from analytical results.
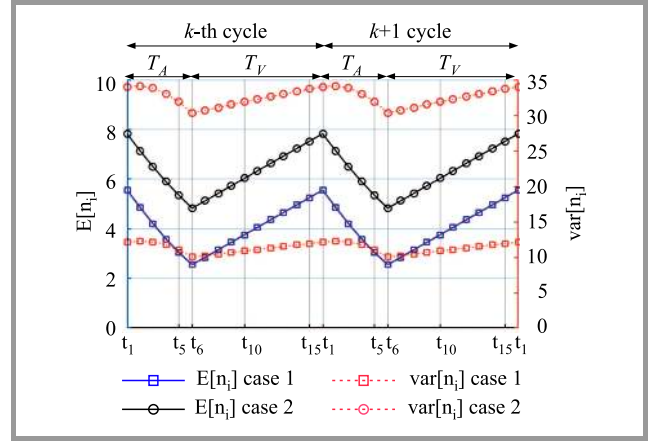

Fig. 8. Mean $E[n_i]$ and variance $var[n_i]$ of packets in the system at times t_i $(i = 1, 2, \dots, N)$.
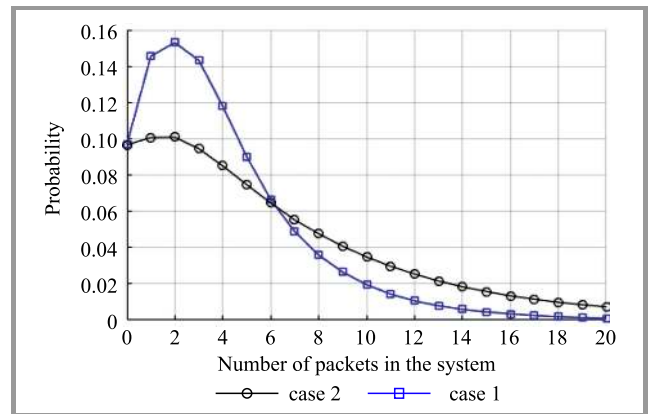

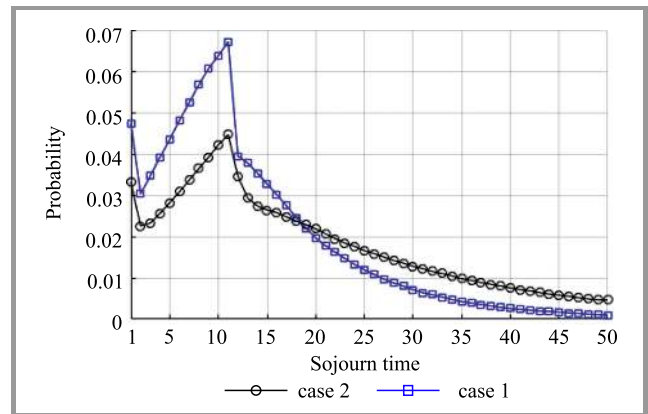Fig. 9. System state distribution for $T_A = 5$ and $T_V = 10$) based on Eq. (9).

Fig. 10. Packet sojourn time distribution for $T_A = 5$ and $T_V = 10$ derived from Eq. (18).

4. Summary

In this paper, we have presented an analysis of a discrete–time multi-queue system with a cycle-based scheduler, in which the packet arrival process may be different for time slots with various positions within the cycle. The analysis of the system focused on calculating system state distribution and, based thereon, packet sojourn time distribution. The formulas presented provide exact solutions. The queueing system refers well to e.g. a system supporting a number of virtual links requiring performance isolation when they share a common physical link as it was implemented in the mentioned IIP System supporting a number of parallel internets.

Appendix A

Proof of Eqs. (10) and (20) identity for the system without vacations

We want to prove that for $n \geq 1$:

$$\Pr\{D = n\} = \frac{\Pr\{X = n\}}{\rho} =$$
$$[\Pr\{X = 0\} + \Pr\{X = 1\}]\frac{\Pr\{n^{Arr} \geq n\}}{\rho} +$$
$$\sum_{m=1}^{n-1} \Pr\{X = m+1\}\frac{\Pr\{n^{Arr} \geq n-m\}}{\rho} .$$

So, actually we need to prove that for $n \geq 1$:

$$\Pr\{X = n\} = [\Pr\{X = 0\} + \Pr\{X = 1\}]\Pr\{n^{Arr} \geq n\} +$$
$$\sum_{m=1}^{n-1} \Pr\{X = m+1\}\Pr\{n^{Arr} \geq n-m\} ,$$

using a mathematical induction.

Base case

For $n = 1$:

$$\Pr\{X = 1\} = (\Pr\{X = 0\} + \Pr\{X = 1\})\Pr\{n^{Arr} \geq 1\} =$$
$$(\Pr\{X = 0\} + \Pr\{X = 1\})(\Pr\{n^{Arr} \geq 0\} - \Pr\{n^{Arr} = 0\}) =$$
$$\Pr\{X = 0\} + \Pr\{X = 1\} - (\Pr\{X = 0\} + \Pr\{X = 1\})\Pr\{n^{Arr} = 0\} =$$
$$\Pr\{X = 1\},$$

since $(\Pr\{X = 0\} + \Pr\{X = 1\})\Pr\{n^{Arr} = 0\} = \Pr\{X = 0\}$ – see Eq. (21).

Inductive step

Let us assume that for an arbitrary $n = k$, $k \geq 1$:

$$\Pr\{X = k\} =$$
$$(\Pr\{X = 0\} + \Pr\{X = 1\})\Pr\{n^{Arr} \geq k\} +$$
$$\sum_{m=1}^{k-1} [\Pr\{X = m+1\}\Pr\{n^{Arr} \geq k-m\}]. \quad (23)$$

For $n = k+1$:

$$\Pr\{X = k+1\} =$$
$$\underbrace{(\Pr\{X = 0\} + \Pr\{X = 1\})\Pr\{n^{Arr} \geq k+1\}}_{C1} +$$
$$\underbrace{\sum_{m=1}^{k} [\Pr\{X = m+1\}\Pr\{n^{Arr} \geq k-m+1\}]}_{C2}$$

Let us rewrite particular components. The first one ($C1$):

$$C1 = (\Pr\{X = 0\} + \Pr\{X = 1\})\Pr\{n^{Arr} \geq k+1\} =$$
$$(\Pr\{X = 0\} + \Pr\{X = 1\})(\Pr\{n^{Arr} \geq k\} - \Pr\{n^{Arr} = k\}) =$$
$$\underbrace{(\Pr\{X = 0\} + \Pr\{X = 1\})\Pr\{n^{Arr} \geq k\}}_{C11} -$$
$$\underbrace{(\Pr\{X = 0\} + \Pr\{X = 1\})\Pr\{n^{Arr} = k\}}_{C12}$$

The $C2$:

$$C2 = \sum_{m=1}^{k} [\Pr\{X = m+1\}\Pr\{n^{Arr} \geq k-m+1\}] =$$
$$\sum_{m=1}^{k} [\Pr\{X = m+1\}\Pr\{n^{Arr} \geq k-m\}] -$$
$$\underbrace{\sum_{m=1}^{k} [\Pr\{X = m+1\}\Pr\{n^{Arr} = k-m\}]}_{C22} =$$
$$\underbrace{\sum_{m=1}^{k-1} [\Pr\{X = m+1\}\Pr\{n^{Arr} \geq k-m\}]}_{C21} +$$
$$\Pr\{X = k+1\} - C22$$

Finally:

$$\Pr\{X = k+1\} = C1 + C2 =$$
$$C11 - C12 + C21 + \Pr\{X = k+1\} - C22 =$$
$$(C11 + C21) - (C12 + C22) + \Pr\{X = k+1\} =$$
$$\Pr\{X = k+1\},$$

since, from Eq. (23): $C11 + C21 = \Pr\{X = k\}$ and also from Eq. (21): $C12 + C22 = \Pr\{X = k\}$.

Since both the base case and the inductive step have been proved as true, by mathematical induction the statement holds for all $n \geq 1$, Q.E.D.

References

[1] W. Burakowski, "Virtualized network infrastructure supporting co existence of Parallel Internets", in *Proc. 13th ACIS Int. Conf. on Software Engin., Artif. Intell., Network. and Parall./Distrib. Compu. SNPD 2012*, Kyoto, Japan, 2012 (DOI: 10.1109/SNPD.2012.67).

[2] B. Vinck and H. Brunnel, "Relationships between delay and buffer contents in ATM queues", *Electron. Lett.*, vol. 31, no. 12, 1995 (DOI: 10.1049/el:19950662).

[3] N. P. Dellaert, *Production to Order. Models and Rules for Production Planning*. Berlin Heidelberg: Springer, 1988 (ISBN: 9783540513094).

[4] M. J. A. Eenige, *Queueing Systems with Periodic Service*. TU Eindhoven, Netherlands, 1996 (ISBN: 9038603487).

[5] H. Takagi, *Queueing Analysis: Vacation and Priority Systems*, pt. 1. Amsterdam: North-Holland, 1991 (ISBN: 9780444817709).

[6] N. Tian and Z. G. Zhang, *Vacation Queueing Models – Theory and Applications*. New York: Springer, 2006 (ISBN: 9780387337210).

[7] B. T. Doshi, "Queueing systems with vacation: A survey", *Queueing Systems*, vol. 1, pp. 29–66, 1986 (DOI: 10.1007/BF01149327).

[8] J. C. Ke, C. H. Wu, and Z. G. Zhang, "Recent developments in vacations queueing models: A short survey", *Int. J. of Oper. Res.*, vol. 7, no. 4, pp. 3–8, 2010 [Online]. Available: http://www.orstw.org.tw/ijor/vol7no4/2-Vol.%207,%20No.4%20pp.3-8.pdf

[9] D. R. McNeil, "A solution to the fixed-cycle traffic light problem for compound Poisson arrivals", *J. of Appl. Probab.*, vol. 5, no. 3, 1968, pp. 624–635 (DOI: 10.2307/3211926).

[10] A. Chydzinski and B. Adamczyk: "Analysis of scheduler for virtualization of links with performance isolation", *Appl. Mathem. & Inform. Sci.*, vol. 8, no. 6, pp. 2653–2666, 2014 (DOI:10.12785/amis/080601).

[11] M. Sosnowski and W. Burakowski. "Analysis of the system with vacations under Poissonian input stream and constant service times", *J. of Telecommun. and Inform. Technol.*, no. 3, 2013 [Online]. Available: https://www.il-pib.pl/czasopisma/JTIT/2013/3/3.pdf

[12] W. Burakowski and M. Sosnowski, "On cycle based schedulers with time alternating priorities", in *Proc. of 27th Int. Telecommun. Netw. and Appl. Conf. ITNAC 2017*, Melbourne, Australia, 2017, pp. 1–6 (DOI: 10.1109/ATNAC.2017.8215377).

Wojciech Burakowski works at the Institute of Telecommunications, Warsaw University of Technology and at the National Institute of Telecommunications and is a member of the Architectures and Applications for the Internet Team. He has participated, since 1990, in several COST and EU Framework Projects. He is a member of the Telecommunications Section of the Polish Academy of Sciences. He was a chairman and a member of many technical program committees at national and international conferences. He is the author or co-author of approximately 250 papers published in books, international and national journals and conference proceedings. Burakowski is also the author of approximately 80 technical reports. His research areas include network techniques for the Internet (ATM, IP QoS, Future Internet), heterogeneous networks (fixed and wireless), network architectures, clouds (including MEC), traffic control, simulation techniques as well as network mechanisms and algorithms. His latest area of interest is in designing wide area research networks for 5G.

https://orcid.org/0000-0002-8486-8004
E-mail: W.Burakowski@il-pib.pl
National Institute of Telecommunications
Szachowa 1
04-894 Warsaw, Poland

Maciej Sosnowski received his M.Sc. degree in Telecommunications from Warsaw University of Technology in 2012. He is a member of the Internet Technologies and Applications Department at the National Institute of Telecommunications, Poland, and concurrently holds a research position within the Architectures and Applications for the Internet (AAI) Team, operating at the Institute of Telecommunications, Warsaw University of Technology. His research interest lies in the areas of queuing theory, virtualization techniques, Internet of Things, cloud computing, multi-access edge computing, and 5G networks.

https://orcid.org/0000-0003-0563-1111
E-mail: M.Sosnowski3@il-pib.pl
National Institute of Telecommunications
Szachowa 1
04-894 Warsaw, Poland