Mathematical and Computer Modelling 57 (2013) 782–792

Contents lists available at SciVerse ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Discrete-time queueing systems with Markovian preemptive vacations

Dieter Fiems *, Herwig Bruneel

SMACS Research Group, Department TELIN, Ghent University, St-Pietersnieuwstraat 41, 9000 Gent, Belgium

ARTICLE INFO

Article history:
Received 3 May 2011
Received in revised form 18 August 2012
Accepted 1 September 2012

Keywords:
Queueing theory
Server vacations
Server breakdowns
Priority queue
CSMA/CD

ABSTRACT

In this contribution we investigate discrete-time queueing systems with vacations. A framework is constructed that allows for studying numerous different vacation systems, including a.o. classical vacation systems like the exhaustive and limited vacation systems as well as queueing systems with service interruptions. Using a probability generating functions approach, we obtain steady-state performance measures such as moments of queue content at different epochs and of customer delay. The usefulness of vacation models in teletraffic is then illustrated by means of some more practical applications (priority queueing, CSMA/CD).

1. Introduction

Queueing systems with vacations [1–3] have proven to be a useful abstraction in modelling unreliability of servers and in modelling systems where service resources are shared between classes of customers. Typical examples of the former class of applications include repair/maintenance models [4] and ARQ systems [5]. Priority queueing models [1,6] and polling models [7,8] are examples of the latter class.

In this contribution we consider the discrete-time $\mathrm{Geo}^X/G/1$ queue subjected to vacations. The vacation process is Markovian and may also depend on the system state: the probability to leave for a vacation at the end of a slot and the duration of this vacation depends on whether or not a customer receives service during the slot, and if so, whether or not this customer remains in service, ends service leaving behind an empty system, or ends service but leaves behind a non-empty system. As such, vacations can interrupt a customer's service; such vacations are sometimes referred to as preemptive, in accordance with the terminology of priority queueing systems. We therefore consider three different operation modes to cope with these interruptions: the customer resumes its service after the interruption, the customer repeats its service or the customer repeats its service with a possibly different (resampled) service time.

The model under consideration can capture behaviour of a number of "classical" vacation models – including the exhaustive vacation system with single and multiple vacations and number- and time-limited vacation systems – as well as of systems with a preemptive independent vacation process. Classical vacation models are extensively treated in Takagi's excellent monographs on continuous-time [1] and discrete-time [9] queueing theory. More recent results are also summarised by Tian and Zhang [10]. Systems with a preemptive independent vacation process are surveyed here. Such vacation models are often referred to as systems with server interruptions or server breakdowns. The availability of the server can then be modelled as an on–off process as server availability alternates between being on and being off.

We first focus on continuous-time models. According to Ibe and Trivedi [11], White and Christie [12] were the first to study queues with interruptions. They consider a continuous-time $M/M/1$ queueing system where the vacation process is modelled as an on–off process with exponentially distributed on- and off-periods. Generally distributed service times

* Corresponding author. Tel.: +32 9 2648902; fax: +32 9 2644295.
E-mail addresses: dieter.fiems@ugent.be, df@telin.UGent.be (D. Fiems), hb@telin.UGent.be (H. Bruneel).

0895-7177/$ – see front matter
doi:10.1016/j.mcm.2012.09.003

and off-periods are considered by Avi-Itzhak and Naor [13] and also by Thiruvengadam [14]. These authors consider exponentially distributed on-periods as opposed to Federgruen and Green [15], who consider phase-type on-periods. Van Dijk [16] provides an approximate analysis of a system with exponentially distributed service times but with generally distributed on- and off-periods whereas Takine and Sengupta [17] study a vacation queueing system in a Markov-modulated environment. The latter authors also allow correlation in the arrival process. Queues with interruptions are also studied outside the framework of single-server first-come-first-served queues. Ke et al. [18] consider a Markovian multi-server queueing system with server interruptions, show that the queueing process can be described by a quasi-birth–death (QBD) process and numerically solve the QBD process. Choudhury and Ke [19] assess performance of a retrial queue with server interruptions by means of a pgf approach. Further, a processor sharing queueing system with exponentially distributed on-periods and generally distributed vacation periods is studied by Núñez Queija [20]. All these contributions assume that customers resume service after the interruption. Gaver Jr. [21] also considers the case where service is either repeated or repeated and resampled after the interruption. The latter operation mode is also studied by Ibe and Trivedi [11] for a two station polling system and by Krishnamoorthy et al. [22] for a queue with a Markov arrival process and phase-type service times.

Research on discrete-time queueing systems with service interruptions started later. Early contributions include those by Hsu [23] and Heines [24]. Both authors treat the single server system with Bernoulli server vacations and a Poisson arrival process. The former considers queue content at random slot boundaries whereas the latter considers queue content at service completion times. A single server system with an independent arrival process and a correlated on/off server vacation process is treated by Bruneel [25], by Yang and Mark [26] and by Woodside and Ho [27]. Yang and Mark [26] and Woodside and Ho [27] model the on- and off-periods as two series of independent shifted geometric random variables, whereas Bruneel [25] assumes that the series of consecutive on-periods as well as the series of consecutive off-periods share a common general distribution. The only restriction in the latter contribution is that the common probability generating function of the on-periods must be rational. Alternatively, correlation in the vacation process is captured by means of a Markovian process by Lee [28].

Georganas [29] and Bruneel [30] treat multi-server systems with independent customer arrival and server vacation processes. The latter extends the former in the sense that it does not assume that all servers are either available or on vacation simultaneously. The delay analysis of the latter system is presented by Laevens and Bruneel [31]. Bruneel [32] also considers a multi-server system with a correlated vacation process. Here, the vacation process is modelled as an on/off process (geometrical on-periods). The numbers of available servers during the consecutive on-slots constitute a series of independent and identically distributed non-negative random variables whereas no servers are available during off-periods.

Some contributions also allow a certain degree of correlation in the arrival process. Bruneel [33] assumes that both arrival and vacation processes are on/off processes with geometric on- and off-periods. A stochastic number of customers enters the system during arrival-on periods, whereas no customers arrive in the system during arrival-off periods. The vacation process is similar to the one analysed by Yang and Mark [26] in the case of uncorrelated arrivals. This vacation process is also considered by Ali et al. [34] and by Kamoun [35]. These authors however assume that customer arrivals come from a superposition of two-state Markovian on–off sources [34] or from a train-arrival process [35].

All the former discrete-time queueing models have fixed customer service times of a single slot in common. Queueing systems where customers have fixed multiple-slot and generally distributed service times are considered by Inghelbrecht et al. [36] and Fiems et al. [6,37,38] respectively. The vacation process is either a Bernoulli [38], a two-state Markovian [36,37] or a renewal process [6]. The combination of multiple-slot service times and vacations implies that service of a customer can be interrupted. The service may then continue [6,36–38] or repeat the service with the same [6,36–38] or a different [38] service time after the interruption. Service may also be repeated partially [6,38] or "delayed" after the interruption [6]. In the latter case, service continues during the vacations but is repeated until the customer receives service without vacations. Finally, as interruptions and service repetitions do render the queueing systems non-work conserving, Morozov and Fiems [39] consider stability of a discrete-time queueing system with server interruptions and resampling after the interruption under the more general setting of generally distributed on-, off- and interarrival times.

The outline of the remainder of this contribution is as follows. In the next section, the model under consideration is described in detail. The analysis is then presented in Sections 3–5. In Section 3, we derive expressions for the probability generating functions of the "effective service times" of customers. The effective service time approach allows us to present a unified queueing analysis for all modes under consideration. The probability generating function of the queue content and customer delay are derived in Sections 4 and 5 respectively. In Section 6, we relate our model to some existing vacation models, whereas some teletraffic applications are presented in Section 7. Finally, conclusions are drawn in Section 8.

2. Mathematical model

We consider a discrete-time queueing system. i.e., time is divided into fixed length intervals or slots. During the consecutive slots, customers arrive in the system, are stored in an infinite capacity buffer and are served on a first-come-first-served basis. The numbers of customers that arrive during the consecutive slots are modelled as a series of independent and identically distributed (i.i.d.) non-negative random variables with common probability mass function (pmf) $e(n)$ $(n \geq 0)$ and corresponding common probability generating function (pgf) $E(z)$. One easily verifies that this arrival process corresponds to a batch arrival process with geometrically distributed inter-arrival times.

Table 1
Symbols used to indicate the different types of vacations.

Symbol	Vacation type
a	Customer that remains in service
b	Customer that leaves behind a non-empty queue
c	Customer that leaves behind an empty queue
d	No customer

Service of customers is synchronized on slot boundaries, implying that a customer cannot start service before the end of its arrival slot. Further, service times – expressed as an integer number of slots – of the consecutive customers are modelled as a series of i.i.d. positive random variables with common pmf $s(n)$ $(n > 0)$ and corresponding pgf $S(z)$. Additionally, we assume that the customer service times are bounded by some maximal value $S_{\max}$. The latter assumption will be relaxed where possible (see further).

We consider a single server system. However, the server is not always available. During each slot where the server is available (A-slot), the vacation process is in one out of N possible states, say state 1 to N. The vacation process is then specified by the following probabilities.

- Given that the vacation process is in state i in a particular A-slot and given that a customer is in service and does not end service during this slot, the server takes a vacation of n $(n \geq 0)$ slots and the vacation process goes to state j after this vacation with probability $b_{ij}^{a}(n)$.
- Similarly, given that the vacation process is in state i in a particular A-slot and given that a customer ends service in this slot and that the system is non-empty after departure of this customer, the server takes a vacation of n $(n \geq 0)$ slots and the vacation process goes to state j after this vacation with probability $b_{ij}^{b}(n)$.
- Also, given that the vacation process is in state i in a particular A-slot and given that a customer ends service in this slot and that the system is empty after departure of this customer, the server takes a vacation of n $(n \geq 0)$ slots and the vacation process goes to state j after this vacation with probability $b_{ij}^{c}(n)$.
- Finally, given that the vacation process is in state i in a particular A-slot and given that there are no customers in the system at the beginning of this slot, the server takes a vacation of n $(n \geq 0)$ slots and the vacation process goes to state j after this vacation with probability $b_{ij}^{d}(n)$.

For convenience and further reference, the meaning of the superscripts a to d above is also summarised in Table 1. One should note that zero-length vacation periods are allowed. In that case, the server is available during the next slot.

For further use, we define the partial probability generating functions $B_{ij}^{k}(z)$, corresponding to the probabilities $b_{ij}^{k}(n)$, for $k \in \{a, \ldots, d\}$,

$$B_{ij}^{k}(z) = \sum_{n=0}^{\infty} b_{ij}^{k}(n)\, z^{n}, \quad |z| < 1 \tag{1}$$

which we collect in $N \times N$ matrices,

$$\mathbf{B}^{k}(z) = \left[B_{ij}^{k}(z)\right]_{i,j=1\ldots N}. \tag{2}$$

As the probabilities $b_{ij}^{k}(n)$ $(n \geq 0)$ are completely specified by the corresponding probability generating functions, the interruption process is therefore also characterized by the $N \times N$ matrices $\mathbf{B}^{k}(z)$, $k \in \{a, \ldots, d\}$.

Clearly, the presence of multiple-slot service times and vacations imply that a vacation can start while a customer receives service. We consider following three operation modes to handle interrupted service. In the *continue after interruption* (CAI) mode, the customer resumes service after the vacation, in the *repeat after interruption* (RAI) mode, the customer has to start all over. This is also the case for the *repeat after interruption with resampling* (RAI,wr) mode. However, in the latter case, service times are resampled after each interruption.

3. Effective service times

Let effective service time of a customer denote the number of slots it effectively takes to serve a particular customer. The effective service time of a customer is defined as the number of slots between the beginning of the slot during which the customer receives service for the first time and the end of the slot in which the customer leaves the system. The effective service times therefore include possible vacation time (service interruptions) and in case of the RAI or RAI,wr mode also lost service slots. Notice that the former definition implies that the server is always available during both the first and the last slot of a customer's effective service time.

3.1. Continue after interruption

We first consider CAI operation. Let $t_{ij}(n|k)$ denote the probability that the effective service time of a customer takes n slots and that the server is in state j during the last slot of the effective service time, given that the server is available and the vacation process is in state i during the first slot of the effective service time, and given that this customer needs k slots of service. For further use, we also define $t_{ij}(n)$ as the same probability without conditioning on the service time.

Conditioning on the length of the vacation that is taken after the first effective service time slot and on the state of the vacation process after this vacation, we get for $1 < k \leq S_{\max}$ and for $i, j \in \{1 \dots N\}$,

$$t_{ij}(n|k) = \sum_{l=1}^{N} \sum_{m=0}^{n-k} t_{lj}(n-m-1|k-1)\, b_{il}^{a}(m), \tag{3}$$

for $n \geq k$ whereas the former probability equals 0 for $n < k$. The former equation holds as, from the vantage point of the server, there is no difference between serving the remaining service time of a customer and serving a new customer with service time equal to that remaining service time.

Let $T_{ij}(z|k)$ denote the partial conditional pgf corresponding to the preceding probabilities, some standard z-transform manipulations then yield,

$$T_{ij}(z|k) \triangleq \sum_{n=k}^{\infty} t_{ij}(n|k)\, z^n = z \sum_{l=1}^{N} T_{lj}(z|k-1)\, B_{il}^{a}(z), \tag{4}$$

for $i, j \in \{1 \dots N\}$ and $1 < k \leq S_{\max}$. If the customer only needs a single slot of service, then its effective service time equals one slot as well by definition. Therefore, we find following probabilities: $t_{ij}(n|1) = \delta_{i-j}\, \delta_{n-1}$, for all n and $i, j \in \{1 \dots N\}$. The corresponding pgf's equal $T_{ij}(z|1) = z\, \delta_{i-j}$ for $i, j \in \{1 \dots N\}$. Here, δ_n denotes the Kronecker delta function, i.e., δ_n equals 1 for $n = 0$ and 0 elsewhere. Note that in this particular case, first and last slot of the effective service time are the same.

For ease of notation, let $\mathbf{T}(z|k)$ denote the $N \times N$ matrix with elements $T_{ij}(z|k)$, Eq. (4) then yields,

$$\mathbf{T}(z|k) = z\, \mathbf{B}^{a}(z)\, \mathbf{T}(z|k-1), \tag{5}$$

for $k > 1$ and further $\mathbf{T}(z|1) = z\, \mathbf{I}_N$. Here $\mathbf{I}_N$ denotes the $N \times N$ unity matrix. Combining the former, we get,

$$\mathbf{T}(z|k) = z\, \left(z\, \mathbf{B}^{a}(z)\right)^{k-1}. \tag{6}$$

Summation over all possible service times with respect to the service time distribution then yields the effective service time matrix $\mathbf{T}(z)$,

$$\mathbf{T}(z) \triangleq \left[\sum_{n=1}^{\infty} t_{ij}(n)\, z^n\right]_{i,j=1\dots N} = z \sum_{k=1}^{S_{\max}} s(k)\, \left(z\, \mathbf{B}^{a}(z)\right)^{k-1}. \tag{7}$$

The matrix $\mathbf{T}(z)$ will be used in Sections 4 and 5.

3.2. Repeat after interruption with resampling

For RAI,wr we proceed similarly. Let $t_{ij}(n|k)$ denote the probability that the effective service time of a customer equals n slots and that the server is available and the vacation process in state j during the last slot of this effective service time, given that the customer needs k slots of service and that the server is available and the vacation process in state i during the first slot of the effective service time. Similarly, let $t_{ij}(n)$ denote the former probability without conditioning on the customer service time.

Conditioning on the length of the vacation taken after the first effective service slot and the state of the vacation process after this vacation we get for $1 < k \leq S_{\max}$ and for $i, j \in \{1 \dots N\}$,

$$t_{ij}(n|k) = \sum_{l=1}^{N} t_{lj}(n-1|k-1)\, b_{il}^{a}(0) + \sum_{l=1}^{N} \sum_{m=1}^{n-2} t_{lj}(n-m-1)\, b_{il}^{a}(m), \tag{8}$$

for $n > 1$. The former holds, as from the vantage point of the server, there is no difference between serving a customer another time with a newly sampled service time and serving a new customer. As for CAI, the effective service time takes 1 slot if the service time takes 1 slot as well. In this case, the first slot of the effective service time is also the last. Therefore, we find $t_{ij}(1|k) = \delta_{k-1}\delta_{i-j}$.

Let $T_{ij}(z|k)$ and $T_{ij}(z)$ denote the (partial conditional) pgf's corresponding to $t_{ij}(n|k)$ and $t_{ij}(n)$ respectively, then, standard z-transform manipulations yield

$$T_{ij}(z|k) = \sum_{l=1}^{N} z\, T_{lj}(z|k-1)\, B_{il}^{a}(0) + \sum_{l=1}^{N} z\, T_{lj}(z)\, (B_{il}^{a}(z) - B_{il}^{a}(0)), \tag{9}$$

for $k > 1$ and $T_{ij}(z|1) = \delta_{i-j}z$. In the same way as for CAI, the former translates in the following matrix equation,

$$\mathbf{T}(z|k) = z\,\mathbf{B}^a(0)\,\mathbf{T}(z|k-1) + z\,(\mathbf{B}^a(z) - \mathbf{B}^a(0))\,\mathbf{T}(z), \tag{10}$$

for $k > 1$ and $\mathbf{T}(z|1) = z\,\mathbf{I}_N$. Combining this with successively applying Eq. (10) then yields,

$$\mathbf{T}(z|k) = \mathbf{\Omega}_k(z) + \mathbf{\Theta}_k(z)(\mathbf{B}^a(z) - \mathbf{B}^a(0))\,\mathbf{T}(z), \tag{11}$$

where $\mathbf{\Omega}_k(z)$ and $\mathbf{\Theta}_k(z)$ are defined as,

$$\mathbf{\Omega}_k(z) = z\,\left(z\,\mathbf{B}^a(0)\right)^{k-1}, \tag{12}$$

and,

$$\mathbf{\Theta}_k(z) = (z\,\mathbf{B}^a(0) - \mathbf{I}_N)^{-1}\,((z\,\mathbf{B}^a(0))^{k-1} - \mathbf{I}_N)\,z, \tag{13}$$

respectively. Summation over all possible service times with respect to the service time distribution then yields,

$$\mathbf{T}(z) = \mathbf{\Omega}(z) + \mathbf{\Theta}(z)\,(\mathbf{B}^a(z) - \mathbf{B}^a(0))\,\mathbf{T}(z). \tag{14}$$

Here $\mathbf{\Omega}(z)$ and $\mathbf{\Theta}(z)$ are given by,

$$\mathbf{\Omega}(z) = \sum_{k=1}^{S_{\max}} s(k)\,\mathbf{\Omega}_k(z), \tag{15}$$

and,

$$\mathbf{\Theta}(z) = \sum_{k=1}^{S_{\max}} s(k)\,\mathbf{\Theta}_k(z), \tag{16}$$

respectively. As for CAI, the matrix $\mathbf{T}(z)$ will be used in the queueing analysis; see Sections 4 and 5.

3.3. Repeat after interruption

For RAI, we may proceed in the same way as we did for CAI and RAI,wr. However it is easier to base our analysis on the obtained results for RAI,wr. To this end, notice that RAI and RAI,wr operate equally when the service times are deterministic. This is observation is immediate as resampling will always result in the same (deterministic) service time. As the service time never changes for RAI, the pmf (and pgf), conditioned on the service time equals the pmf (and pgf), assuming a deterministic service time. These observations show that the pgf of the effective service time of a customer given that its service time is k slots, can be obtained by substituting $s(k) = \delta_k$ – the pmf corresponding to fixed length service times of k slots – in Eqs. (14)–(16),

$$\mathbf{T}(z|k) = (\mathbf{I}_N - \mathbf{\Theta}_k(z)\,(\mathbf{B}^a(z) - \mathbf{B}^a(0)))^{-1}\,\mathbf{\Omega}_k(z), \tag{17}$$

where the matrices $\mathbf{\Omega}_k(z)$ and $\mathbf{\Theta}_k(z)$ are defined in expressions (12) and (13) respectively.

Summation over all possible service times with respect to their probabilities then yields the following expression for the matrix $\mathbf{T}(z)$ in case of RAI operation,

$$\mathbf{T}(z) = \sum_{k=1}^{S_{\max}} s(k)\,\left(\mathbf{I}_N - \mathbf{\Theta}_k(z)\,(\mathbf{B}^a(z) - \mathbf{B}^a(0))\right)^{-1}\,\mathbf{\Omega}_k(z). \tag{18}$$

The matrix $\mathbf{T}(z)$ will be used in the following sections.

4. Queue content

We first consider the queue content at departure epochs, i.e., at the beginning of a slot following a slot where a customer leaves the system. Let U_k denote the queue content at the departure epoch of the kth customer and let Q_k denote the state of the vacation process during the slot where this customer leaves the system. Further, let $U_k(z, j)$ denote the partial pgf corresponding to the queue content at the kth departure epoch and the corresponding state of the vacation process,

$$U_k(z, j) \triangleq \mathrm{E}\left[z^{U_k}|Q_k = j\right]\Pr\left[Q_k = j\right], \tag{19}$$

for $j \in \{1 \dots N\}$. For ease of notation, we let $\mathbf{U}_k(z)$ denote the row vector with elements $U_k(z, j)$,

$$\mathbf{U}_k(z) = \left[U_k(z, 1) \dots U_k(z, N)\right]. \tag{20}$$

We now relate the queue content at the kth and the $(k+1)$st departure epoch. Given that the queue is empty after departure of the kth customer, a vacation characterized by the matrix $\mathbf{B}^c(z)$ is taken, followed by vacations characterized by the matrix $\mathbf{B}^d(z)$ until there is at least one customer in the queue upon returning from a vacation. A customer is then served which leaves the system at the $(k+1)$st departure epoch. On the other hand, given that the kth customer leaves a non-empty system behind, a vacation characterized by the matrix $\mathbf{B}^b(z)$ is taken, and the $(k+1)$th customer starts service immediately thereafter. Using some standard z-transform and matrix manipulations, one retrieves following relation between the vectors $\mathbf{U}_{k+1}(z)$ and $\mathbf{U}_k(z)$,

$$\mathbf{U}_{k+1}(z) = (\mathbf{U}_k(z) - \mathbf{U}_k(0))\,\mathbf{B}^b(E(z))\,\frac{1}{z}\,\mathbf{T}(E(z)) + \mathbf{U}_k(0)\,\left(\mathbf{B}^c(E(z)) - \mathbf{B}^c(e_0)\right)\,\frac{1}{z}\,\mathbf{T}(E(z)) + \mathbf{U}_k(0)\,\mathbf{B}^c(e_0)\,\mathbf{\Lambda}(z)\,\frac{1}{z}\,\mathbf{T}(E(z)), \tag{21}$$

where we introduced

$$\mathbf{\Lambda}(z) = \sum_{i=0}^{\infty} \tilde{\mathbf{B}}^d(e_0)^i\,\left(\tilde{\mathbf{B}}^d(E(z)) - \tilde{\mathbf{B}}^d(e_0)\right) = \left(\mathbf{I}_N - \tilde{\mathbf{B}}^d(e_0)\right)^{-1}\left(\tilde{\mathbf{B}}^d(E(z)) - \tilde{\mathbf{B}}^d(e_0)\right) \tag{22}$$

and

$$\tilde{\mathbf{B}}^k(z) = z\mathbf{B}^k(z) \tag{23}$$

for $k = a, \ldots, d$ to simplify notation. The consecutive terms on the right-hand side of (21) correspond to (i) the case that there are customers in the queue upon departure of the kth customer, (ii) the case that there are no customers upon departure of the kth customer, but that there are customers in queue after the vacation which is taken upon departure of this customer and (iii) the case that there are still no customers in the system after the formerly mentioned vacation.

Under the assumption that the system under consideration reaches equilibrium, let $\mathbf{U}(z)$ denote the vector of partial pgf's of the queue content at departure times in equilibrium. The former equation then easily yields,

$$\mathbf{U}(z)\mathbf{\Gamma}_1(z) - \mathbf{U}(0)\mathbf{\Gamma}_2(z) = 0, \tag{24}$$

with,

$$\mathbf{\Gamma}_1(z) = z\,\mathbf{I}_N - \mathbf{B}^b(E(z))\,\mathbf{T}(E(z)), \tag{25}$$

$$\mathbf{\Gamma}_2(z) = \left[\mathbf{B}^c(E(z)) - \mathbf{B}^b(E(z))\right]\,\mathbf{T}(E(z)) + \mathbf{B}^c(e_0)\,(\mathbf{\Lambda}(z) - \mathbf{I}_N)\,\mathbf{T}(E(z)). \tag{26}$$

Expression (24) shows that the vacation system under consideration on departure epochs is of $M/G/1$ type. One may therefore retrieve the unknown vector $\mathbf{U}(0)$ as follows [40]:

1. Find all points ξ_j in $|z| < 1$ where $\mathbf{\Gamma}_1(z)$ is singular. For each ξ_j, retrieve a non-zero column vector Ξ_j such that $\mathbf{\Gamma}_1(\xi_j)\Xi_j = 0$.
2. For all j, plugging ξ_j into Eq. (24) and multiplying on the right by the corresponding Ξ_j, leads to a linear equation for the unknown vector $\mathbf{U}(0)$.
3. The normalisation condition $\mathbf{U}(1)\,\mathbf{e}^T = 1$ leads to another linear equation for the unknown vector $\mathbf{U}(0)$. Here $\mathbf{e}^T$ denotes the $N \times 1$ column vector with all elements equal to 1.
4. The unknown vector $\mathbf{U}(0)$ is then retrieved by solving the former set of linear equations.

Once one retrieves $\mathbf{U}(0)$, one may determine $\mathbf{U}(z)$ by means of expression (24)–(26).

We now focus on the pgf's of the buffer content at various epochs in time. Clearly, the pgf of the buffer content at departure epochs $U(z)$ is given by,

$$U(z) = \mathbf{U}(z)\,\mathbf{e}^T. \tag{27}$$

Under the assumption that there are no simultaneous arrivals within slots, the former pgf is also the pgf of the queue content on arrival epochs (see, e.g., [9]). Further the pgf $N(z)$ of the buffer content at random slot boundaries relates to the generating function $U(z)$ of the buffer content at departure epochs as (see e.g., [41]),

$$N(z) = E'(1)\,\frac{1-z}{1-E(z)}\,U(z). \tag{28}$$

The moment generating property of pgf's then allows us to obtain various performance measures such as the mean and the variance of the queue content in equilibrium.

5. Customer delay

Consider a random (tagged) customer and let D denote this customer's delay, U denote the number of customers in the queue upon departure of this customer and $\hat{E}$ denote the number of customers that arrive during the same slot as the tagged customer but that receive service after the tagged customer. These variables then relate as,

$$U = \hat{E} + \sum_{j=1}^{D} E_j, \tag{29}$$

with E_j the number of arrivals during the j-th slot that the customer spends in the queue. This expression follows from the fact that all customers that are in the queue upon departure of the tagged customer, either arrived during this customers delay or in the same slot as but after the tagged customer. Unfortunately, $\hat{E}$ and D are correlated. It is however possible to avoid this correlation problem by means of the "customer batch service" approach [9].

Let "batch" refer to the number of customers that arrive during a random slot where there is at least one customer arrival. Due to the i.i.d. nature of the arrival process, the number of batch arrivals during the consecutive slots constitute a series of independent Bernoulli distributed random variables. There is a batch arrival during a slot with probability $1 - E(0)$. As such, the common pgf of the number of batch arrivals during the consecutive slots is given by,

$$\tilde{E}(z) = E(0) + (1 - E(0))\, z. \tag{30}$$

The batch effective service time is defined as the number of slots between the beginning of the slot where the first customer of the batch receives service for the first time and the end of the slot where the last customer of the batch leaves the system. As such, the batch effective service time equals the sum of the effective service times of all customers in the batch and of the vacations that may take place between serving customers of the batch. Let $\tilde{\mathbf{T}}(z)$ denote the matrix of the partial (the state of the vacation process during the last slot of the batch effective service time) conditional (the state of the vacation process during the first slot of the batch effective service time) pgf's of the batch effective service times, then,

$$\tilde{\mathbf{T}}(z) = \sum_{j=1}^{\infty} \frac{e(j)}{1 - e(0)} \left(\mathbf{T}(z)\,\mathbf{B}^b(z)\right)^{j-1} \mathbf{T}(z). \tag{31}$$

We can now obtain the pgf of the queue content (in terms of batches) at batch departure times by replacing effective service time by batch effective service time and customer arrivals by batch customer arrivals in the expressions of the former section. This then leads to the vector $\tilde{\mathbf{U}}(z)$, the vector of the partial pgf's of the queue content in terms of batches at batch departure times. Notice that $\tilde{\mathbf{U}}(0) = \mathbf{U}(0)$ as no batches in the queue implies no customers in the queue and vice versa.

Let batch waiting time denote the number of slots between the end of the slot where the batch arrives in the queue and the beginning of the slot where the first customer of the batch is served. All batches in the queue upon departure of a batch either arrive during the batch waiting time or the batch effective service time of this batch as there is at most one batch arrival during a slot. Let $\tilde{\mathbf{W}}(z)$ denote the vector of the partial (on the state of the vacation process during the batch departure slot) pgf's of the batch waiting time, then,

$$\tilde{\mathbf{U}}(z) = \tilde{\mathbf{W}}(\tilde{E}(z))\,\tilde{\mathbf{T}}(\tilde{E}(z)). \tag{32}$$

Clearly, the former expression allows us to determine $\tilde{\mathbf{W}}(z)$ in terms of known variables.

Finally, a tagged customer's delay consists of the batch waiting time of the batch where this customer belongs to, of the effective service times of all customers that arrive during the tagged customer's arrival slot but that are served before the tagged customer, of the effective service time of the tagged customer and of the (possible) vacations between these effective service times. Let $D(z)$ denote the pgf of the delay of a random customer, then,

$$D(z) = \tilde{\mathbf{W}}(z) \sum_{j=0}^{\infty} \dot{e}(j) \left(\mathbf{T}(z)\,\mathbf{B}^b(z)\right)^j \mathbf{T}(z)\mathbf{e}^T, \tag{33}$$

where $\dot{e}(j)$ denotes the probability that there are j customers that arrive in a random customer's arrival slot but that are served before this random customer. This probability is given by [41],

$$\dot{e}(j) = \frac{1}{E'(1)} \sum_{k=j}^{\infty} e(k). \tag{34}$$

The moment generating property of pgf's then allows us to determine performance measures such as mean and variance of the customer delay. Notice that the infinite sums in Eqs. (31) and (33) do not pose problems as we can plug in the eigenvalue decomposition of the matrix $(\mathbf{T}(1)\,\mathbf{B}^b(1))$ after the derivation and evaluation in $z = 1$ of these expressions.

6. Special cases

As mentioned in the introductory section, our model can capture the behaviour of various existing vacation models. Some are considered here.

6.1. Exhaustive vacation systems

In a system with exhaustive vacations, the server starts a vacation whenever the queue is empty after the departure of a customer. If the queue is still empty upon returning from a vacation, the server either immediately takes another vacation or remains idle until a new customer arrives. One refers to these two policies as the multiple and the single vacation policy respectively.

Under the assumption that the consecutive vacations constitute a series of i.i.d. random variables, behaviour of the $\text{Geo}^X/G/1$ queue with multiple and single vacations can be retrieved using the following 1×1 vacation matrices,

$$\begin{aligned} \mathbf{B}^a(z) &= \mathbf{B}^b(z) = \left[1\right], \\ \mathbf{B}^c(z) &= \left[V(z)\right], \end{aligned} \tag{35}$$

and using either $\mathbf{B}^d(z) = \left[1\right]$ or $\mathbf{B}^d(z) = \left[\frac{V(z)}{z}\right]$, for the single and the multiple vacation system respectively. Here, $V(z)$ denotes the pgf of the consecutive vacations. As there are no vacations during a customer's service, the latter is never interrupted and therefore a customers effective service times equals its service time. As such, there is clearly no need to pose an upper bound for the customer service times. Our results comply with Takagi's results [9, pp. 98 and 132].

Note that the framework at hand allows for various extensions of classical exhaustive vacation systems. For example, the distribution of the vacation period can depend on the number of vacations taken so far. Such vacation systems have been used to study sleep-mode operation in wireless networks including WiMax and LTE [42,43].

6.2. Queueing systems with interruptions

Consider a $\text{Geo}^X/G/1$ system where the single server alternates between available and vacation periods, independently of the rest of the system. In particular, one may consider the case where the consecutive available and vacation periods are modelled by a series of i.i.d. random variables and where the available periods share a common geometrical distribution. It is shown [6] that such a model captures the performance of low-priority traffic in a $\text{Geo}^X/G/1$ preemptive priority queue. One easily verifies that following vacation matrix correspond to the system under consideration,

$$\mathbf{B}^k(z) = \left[\alpha + (1-\alpha)\, V(z)\right], \tag{36}$$

for $k = a, \ldots, d$. Here α denotes the probability that an available period continues during the next slot and $V(z)$ denotes the common pgf of the vacation periods. The former results comply with those presented in [6]. One may further verify, that in this particular case we do not have to pose an upper bound for the service times.

6.3. Non-preemptive time-limited systems

In time-limited systems, the server takes a vacation whenever there are either no more customers to be served or whenever a timer (restarted after each vacation) expires. In the case of non-preemptive time-limited vacation systems, the server takes a vacation after finishing service of a customer during which the timer expired. If one assumes a time-limited $\text{Geo}^X/G/1$ system with geometrically distributed timers, one retrieves this system's performance measures using the following matrices:

$$\begin{aligned} \mathbf{B}^a(z) &= \begin{bmatrix} \alpha & 1-\alpha \\ 0 & 1 \end{bmatrix}, \\ \mathbf{B}^b(z) &= \mathbf{B}^c(z) = \begin{bmatrix} \alpha + (1-\alpha)V(z) & 0 \\ V(z) & 0 \end{bmatrix}, \\ \mathbf{B}^d(z) &= \begin{bmatrix} \frac{V(z)}{z} & 0 \\ 1 & 0 \end{bmatrix}. \end{aligned} \tag{37}$$

Here, α denotes the probability that the timer does not expire during a slot and $V(z)$ denotes the common pgf shared by the consecutive (independent) vacations. Note that state 1 corresponds to slots where the timer is active, whereas state 2 corresponds to slots where this is not the case. The results comply with those presented in [44]. Again, one can verify that we do not have to pose an upper bound for the service times in this case.

6.4. E-limited vacation systems

For E-limited vacation systems or exhaustive number-limited vacation systems, the server takes a vacation whenever there are either no more customers to be served or whenever a fixed number N of customers have been served since the last vacation. Our model simplifies to the $\mathrm{Geo}^X/G/1E$-limited multiple vacation system if one considers the following vacation matrices,

$$\begin{aligned}
\mathbf{B}^a(z) &= \mathbf{I}_N, \\
\mathbf{B}^b(z) &= \mathbf{B}^c(z) = \left[\delta_{i-j+1} + \delta_{i-N}\delta_{j-1}V(z)\right]_{i,j=1\ldots N}, \\
\mathbf{B}^d(z) &= \frac{V(z)}{z}\left[\delta_{j-1}\right]_{i,j=1\ldots N}.
\end{aligned} \tag{38}$$

Here $V(z)$ denotes the common pgf of the consecutive (independent) vacation periods. During customer service, the vacation state corresponds to the number of customers that started service since the last vacation. Clearly, we do not have to pose an upper bound for the service times in this case as service is never interrupted. The results comply with those presented in [9, pp. 209–214].

7. Applications

We now shift focus to some more practical applications. In particular, we investigate some priority queueing models and a (simplified) carrier sense multiple access with collision detection (CSMA/CD) protocol.

7.1. Preemptive priority queues

Consider a 2-class preemptive priority queue. Low-priority (LP) packets (customers) are only transmitted (served) when there are no high-priority (HP) packets in the system. Therefore, from the vantage point of LP packets, the transmission channel (the server) leaves for a vacation during the HP busy periods—periods where HP packets are transmitted. For preemptive priority queueing disciplines, transmission of the LP packets is immediately postponed when HP packets arrive. When all HP packets are transmitted, the transmission of the LP packet is either resumed (preemptive resume priority) or repeated with the same (preemptive repeat identical priority) or a different transmission time (preemptive repeat different). See a.o. [6] and the references therein.

Consider in particular the case that the HP packets arrive in an infinite capacity buffer according to a discrete batch Markovian arrival process (DBMAP) and that the transmission times of these packets constitute a series of i.i.d. random variables. In this case, the amount of HP work arriving during the consecutive slots – the total transmission times of all packet arrivals during a slot – constitutes a series of Markov-modulated random variables. Let the matrices $\mathbf{W}_k$ characterise this HP work process:

$$\mathbf{W}_k = \left[\Pr[W = k, Q' = j | Q = i]\right]_{i,j=1\ldots N}. \tag{39}$$

Here W denotes the amount of HP work that arrives during a random slot, whereas Q and Q' ($Q, Q' \in \{1 \ldots N\}$) denote the state of the arrival process during this random slot and the following slot respectively.

We here use the following definition of busy period: The HP busy period starts at the end of a slot where there are no HP packets present in the system and ends at the beginning of the slot where the HP queue is empty for the first time since the beginning of the busy period. Notice that 0-slot busy periods are in accordance with this definition. A 0-slot busy period corresponds to the case where the HP queue remains empty during two consecutive slots.

Consider a random (tagged) slot where the HP queue is empty. The slot following the tagged slot is the next slot where the HP queue is empty if there are no arrivals during the tagged slot. If there are arrivals, the busy period equals the sum of the amount of work that arrives during the tagged slot, augmented with the length of a busy period for each of the units of work that arrive,

$$B = W + \sum_{j=1}^{W} B_j. \tag{40}$$

Here B denotes the busy period corresponding to the tagged slot and the B_j's denote the busy periods corresponding to the units of work that arrived during the tagged slot.

The busy periods only depend on the state of the work process during the slot preceding the busy period. That is, the consecutive busy periods constitute a series of Markov-modulated random variables. Let $\mathbf{B}(z)$ denote matrix of the partial conditional pgf's of the busy periods,

$$\mathbf{B}(z) = \left[B_{ij}(z)\right]_{i,j=1\ldots N}, \tag{41}$$

with,

$$B_{ij}(z) = \sum_{k=0}^{\infty} \Pr[B = k, Q' = j | Q = i]\, z^k. \tag{42}$$

Here Q and Q' denote the state of the work process during the slot preceding and following the busy period respectively.

In view of Eq. (40) and the Markovian nature of the busy period, we find the following functional equation for the matrix $\mathbf{B}(z)$,

$$\mathbf{B}(z) = \sum_{k=0}^{\infty} \mathbf{W}_k\, [z\, \mathbf{B}(z)]^k. \tag{43}$$

The former functional equation allows us to numerically evaluate $\mathbf{B}(z)$ and its derivatives for all z.

Recall that the busy periods are perceived as vacation periods by the LP packets. As such, we may evaluate the performance of the low-priority customers if we assume that all vacation matrices are equal to $\mathbf{B}(z)$:

$$\mathbf{B}^a(z) = \mathbf{B}^b(z) = \mathbf{B}^c(z) = \mathbf{B}^d(z) = \mathbf{B}(z). \tag{44}$$

The CAI operation mode then corresponds to preemptive resume priority queueing, the RAI mode to preemptive repeat identical priority queueing and the RAI,wr mode to preemptive repeat different priority queueing.

7.2. CSMA/CD

Consider a half-duplex bus network with a CSMA/CD based media access control. All network transceivers are equipped with an output buffer to temporarily store outgoing packets. Clearly, from the vantage point of packets from a single transceiver, the output line (the common bus) of the transceiver is unavailable from time to time as other transceivers may gain access to the bus. i.e., the server of the output buffer leaves for a vacation from time to time.

The (simplified) CSMA/CD protocol under consideration operates as follows. Whenever there are no packets to be transmitted, the transceiver under consideration listens to the channel (carrier sense). Upon arrival of packets, the transceiver either starts packet transmission if the channel is available or delays packet transmission until the channel becomes available. Due to transmission delays on the common bus, it is possible that multiple transceivers simultaneously start transmission. This leads to collisions which can be detected by the transceivers (collision detection). Whenever a collision is detected the transceiver first sends out a jam sequence to make sure that all transceivers detect the collision and, after waiting for some random amount of time, retransmits the packet as soon as the channel becomes available. Upon successful transmission, the transceiver relinquishes control of the transmission medium.

We now show how to construct the corresponding vacation process. As in [45], we hereby assume that there is a fixed probability α that another transceiver starts transmission in a slot, given that this transceiver is unaware of other transmissions. Recall that vacations correspond to the time that the channel is unavailable for transmission.

First, consider the vacation process during a packet's transmission. When the transmission starts, collisions may occur as is it possible that other transceivers are not aware of the transmission. After some time however all transceivers are aware of the current transmission and no collisions occur. We therefore introduce two vacation states. State 1 corresponds to the initial phase of the transmission. Another transceiver starts a transmission and causes a collision with probability α. For ease of analysis, we assume that the length of this first phase is geometrically distributed with mean $1/(1-p)$. i.e., the vacation process remains in state 1 during the following slot with probability p. Further, state 2 corresponds to the case that there are no more collisions. Whenever the vacation process enters state 2 during a transmission, it remains in state 2 until the end of the transmission. This leads to the following vacation matrix,

$$\mathbf{B}^a(z) = \begin{bmatrix} p\,\eta(z) & (1-p)\,\eta(z) \\ 0 & 1 \end{bmatrix}, \tag{45}$$

with,

$$\eta(z) = 1 - \alpha + \alpha\, V_1(z). \tag{46}$$

Here $V_1(z)$ is the probability generating function of the period that the transceiver must wait before starting retransmission. This period includes the jam sequence, the random waiting time and the time it takes until the channel is free.

At the end of a transmission or whenever there is no work in the queue, other transceivers may gain access to the channel. As another transceiver starts a transmission with probability α, we find,

$$\mathbf{B}^b(z) = \mathbf{B}^c(z) = \mathbf{B}^d(z) = \begin{bmatrix} 1-\alpha+\alpha\, V_2(z) & 0 \\ 1-\alpha+\alpha\, V_2(z) & 0 \end{bmatrix}. \tag{47}$$

Here $V_2(z)$ denotes the pgf of the number of slots that another transceiver gains access to the channel.

The vacation model may be extended to incorporate more properties of CSMA/CD protocols. e.g. additional states can be defined to incorporate the exponential increase of the back-off timer, to embed the correlation in the channel occupation by the other transceivers or to model the length of the collision period more accurately.

8. Conclusions

We considered a discrete-time $\text{Geo}^X/G/1$ queueing system subjected to correlated vacations and derived an expression for the pgf of the queue content at various epochs in time and of the customer delay. We showed that our model generalizes various existing vacation models and illustrated the usefulness of queues with vacations in teletraffic by means of some more practical applications.

References

[1] H. Takagi, Queueing Analysis; A foundation of Performance Evaluation, volume 1: Vacation and Priority Systems, Part 1, Elsevier Science Publishers, 1991.
[2] B. Doshi, Queueing systems with vacations—a survey, Queueing Systems 1 (1986) 29–66.
[3] D. Fiems, S. DeVuyst, H. Bruneel, The combined gated-exhaustive vacation system in discrete time, Performance Evaluation 49 (1–4) (2002) 227–239.
[4] F. Van der Duyn Schouten, S. Vanneste, Maintenance optimization of a production system with buffer capacity, European Journal of Operational Research 82 (1995) 232–338.
[5] D. Towsley, A statistical analysis of ARQ protocols operating in a nonindependent error environment, IEEE Transactions on Communications 29 (7) (1981) 971–981.
[6] D. Fiems, B. Steyaert, H. Bruneel, Discrete-time queues with generally distributed service times and renewal-type server interruptions, Performance Evaluation 55 (3–4) (2003) 277–298.
[7] H. Takagi, A survey of queueing analysis of polling models, in: Proceedings of the Third IFIP International Conference on Data Communication Systems and Their Performance, Rio de Janeiro, Brazil, June 1987, pp. 22–25.
[8] E. Altman, D. Fiems, Expected waiting time in polling systems with correlated vacations, Queueing Systems 56 (3–4) (2007) 241–253.
[9] H. Takagi, Queueing Analysis; A foundation of Performance Evaluation, volume 3: Discrete-Time Systems, Elsevier Science Publishers, Amsterdam, 1993.
[10] N. Tian, Z.G. Zhang, Vacation Queueing Models, Theory and Applications, Springer, 2006.
[11] O. Ibe, K. Trivedi, Two queues with alternating service and server breakdown, Queueing Systems 7 (3–4) (1990) 253–268.
[12] H. White, L. Christie, Queuing with preemptive priorities or with breakdown, Operations Research 6 (1) (1958) 79–95.
[13] B. Avi-Itzhak, P. Naor, Some queuing problems with the service station subject to breakdown, Operations Research 11 (3) (1963) 303–319.
[14] K. Thiruvengadam, Queuing with breakdowns, Operations Research 11 (1) (1963) 62–71.
[15] A. Federgruen, L. Green, Queueing systems with service interruptions, Operations Research 34 (5) (1986) 752–768.
[16] N. Van Dijk, Simple bounds for queueing systems with breakdowns, Performance Evaluation 8 (2) (1988) 117–128.
[17] T. Takine, B. Sengupta, A single server queue with service interruptions, Queueing Systems 26 (1997) 285–300.
[18] J.C. Ke, C.H. Lin, H.I. Huang, Z.G. Zhang, An algorithmic analysis of multi-server vacation model with service interruptions, Computers & Industrial Engineering 61 (4) (2011) 1302–1308.
[19] G. Choudhury, J.C. Ke, A batch arrival retrial queue with general retrial times under Bernoulli vacation schedule for unreliable server and delaying repair, Applied Mathematical Modelling 36 (1) (2012) 255–269.
[20] R. Núñez Queija, Sojourn times in a processor sharing queue with service interruptions, Queueing Systems 34 (1–4) (2000) 351–386.
[21] D. Gaver Jr., A waiting line with interrupted service, including priorities, Journal of the Royal Statistical Society. Series B 24 (1962) 73–90.
[22] A. Krishnamoorthy, P.K. Pramod, T.G. Deepak, On a queue with interruptions and repeat or resumption of service, Nonlinear Analysis Theory Methods & Applications 71 (12) (2009) E1673–E1683.
[23] J. Hsu, Buffer behavior with Poisson arrival and geometric output processes, IEEE Transactions on Communications 22 (1974) 1940–1941.
[24] T. Heines, Buffer behavior in computer communication systems, IEEE Transactions on Communications 28 (1979) 573–576.
[25] H. Bruneel, A general treatment of discrete-time buffers with one randomly interrupted output line, European Journal of Operational Research 27 (1) (1986) 67–81.
[26] O. Yang, J. Mark, Performance analysis of integrated services on a single server system, Performance Evaluation 11 (1990) 79–92.
[27] C. Woodside, E. Ho, Engineering calculation of overflow probabilities in buffers with Markov-interrupted service, IEEE Transactions on Communications 35 (12) (1987) 1272–1277.
[28] D. Lee, Analysis of a single server queue with semi-Markovian service interruption, Queueing Systems 27 (1–2) (1997) 153–178.
[29] N. Georganas, Buffer behavior with Poisson arrivals and bulk geometric output processes, IEEE Transactions on Communications 24 (8) (1976) 938–940.
[30] H. Bruneel, A general model for the behaviour of infinite buffers with periodic service opportunities, European Journal of Operational Research 16 (1984) 98–106.
[31] K. Laevens, H. Bruneel, Delay analysis for discrete-time queueing systems with multiple randomly interrupted servers, European Journal of Operational Research 85 (1995) 161–177.
[32] H. Bruneel, A discrete-time queueing system with a stochastic number of servers subjected to random interruptions, Opsearch 22 (4) (1985) 215–231.
[33] H. Bruneel, On buffers with stochastic input and output interruptions, International Journal of Electronics and Communications (AEÜ) 38 (4) (1984) 265–271.
[34] M. Ali, X. Zhang, J. Hayes, A discrete-time queueing analysis of the wireless ATM multiplexing system, in: Networking—ICN 2001 (Part I), in: Lecture Notes in Computer Science, vol. 2093, Springer, Colmar, France, 2001, pp. 429–438.
[35] F. Kamoun, Performance evaluation of a queuing system with correlated packet-trains and server interruption, Telecommunication Systems 41 (4) (2009) 267–277.
[36] V. Inghelbrecht, K. Laevens, H. Bruneel, B. Steyaert, Queueing of fixed-length messages in the presence of server interruptions, in: Proceedings Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2k, Vancouver, Canada, July 2000.
[37] D. Fiems, B. Steyaert, H. Bruneel, Analysis of a discrete-time $GI-G-1$ queueing model subjected to bursty interruptions, Computers and Operations Research 30 (1) (2002) 139–153.
[38] D. Fiems, B. Steyaert, H. Bruneel, Randomly interrupted $GI-G-1$ queues, service strategies and stability issues, Annals of Operations Research 112 (2002) 171–183.
[39] E. Morozov, D. Fiems, Stability analysis of multiserver discrete-time queueing systems with renewal-type server interruptions, Performance Evaluation 68 (12) (2011) 1261–1275.
[40] H. Gail, S. Hantler, B. Taylor, Spectral analysis of $M/G/1$ and $G/M/1$ type Markov chains, Advances in Applied Probability 28 (1996) 114–165.
[41] H. Bruneel, B.G. Kim, Discrete-Time Models for Communication Systems Including ATM, Kluwer Academic Publishers, 1993.
[42] K. De Turck, S. De Vuyst, D. Fiems, S. Wittevrongel, Performance analysis of the IEEE 802.16e sleep mode for correlated downlink traffic, Telecommunication Systems 39 (2) (2008) 145–156.
[43] K. De Turck, S. De Vuyst, D. Fiems, S. Wittevrongel, H. Bruneel, Performance analysis of sleep mode mechanisms in the presence of bidirectional traffic, Computer Networks 56 (10) (2012) 2494–2505.
[44] D. Fiems, H. Bruneel, Analysis of a discrete-time queueing system with timed vacations, Queueing Systems 42 (3) (2002) 243–254.
[45] H. Zhai, Y. Kwon, Y. Fang, Performance analysis of IEEE 802.11 MAC protocols in wireless LANs, Wireless Communications and Mobile Computing 4 (2004) 917–931.