The Preemptive Repeat Hybrid Server Interruption Model

Thomas Demoor[1], Dieter Fiems[1], Joris Walraevens[1], and Herwig Bruneel[1]

Ghent University, St. Pietersnieuwstraat 41, 9000 Gent,Belgium
{thdemoor,df,jw,hb}@telin.ugent.be

Abstract. We analyze a discrete-time queueing system with server interruptions and a hybrid preemptive repeat interruption discipline. Such a discipline encapsulates both the preemptive repeat identical and the preemptive repeat different disciplines. By the introduction and analysis of so-called service completion times, we significantly reduce the complexity of the analysis. Our results include a.o. the probability generating functions and moments of queue content and delay. Finally, by means of some numerical examples, we assess how performance measures are affected by the specifics of the interruption discipline.

1 Introduction

In many queueing systems, the server is not continuously available (for all types of customers). Service interruptions may result from repair times after server failures, from planned maintenance periods or from multiple queues sharing a server (priority queues, polling systems). In the latter case, the server is unavailable for a particular queue whenever it serves customers from other queues. Obviously, these service interruptions, often also entitled server vacations or server breakdowns, have a major impact on the operation of a queueing system and cannot be neglected when analyzing this system.

Evidently, the effect of server vacations is most striking when ongoing service of a customer can be interrupted. Different fashions of reengaging service after a service interruption have given rise to several types of vacation models. One speaks of preemptive resume (PR) if the interrupted customer can continue his service, of preemptive repeat identical (PRI) if his service is restarted or of preemptive repeat different (PRD) if his service is restarted with a new service time. The used terminology stems from the priority queueing context. In the literature on machine breakdowns, PRI is simply called preemptive repeat and PRD is called preemptive repeat with resampling. Naturally, repeated service has more impact on system performance than continued service. The current contribution studies a hybrid of PRI and PRD which we have baptized preemptive repeat hybrid (PRH). Here, after each service preemption, service is repeated with a new service time with probability γ or with the same service time with probability $1-\gamma$. Evidently, PRH encapsulates the known preemptive repeat vacation models as setting γ to 0 or 1 yields PRI or PRD respectively.

The proposed model is studied in discrete-time under the assumption that the interruption process is independent of the arrival and service processes allowing separate analysis of the interruption and queueing processes.

Vacation models have been studied for over 50 years both in continuous and discrete time. To the best of our knowledge, they were first studied in connection with priority queueing systems by White and Christie [1]. These authors investigated the $M/M/1$ queueing system with a preemptive resume priority discipline. Their results were later extended to general service times by Avi-Itzhak and Naor [2] and by Thiruvengadam [3]. Gaver Jr. [4] investigated the preemptive repeat identical and preemptive repeat different disciplines in a priority queuing system with batch Poisson arrivals and generally distributed service times. More recently, Fiems, Steyaert and Bruneel [5] considered the discrete-time $M^X/G/1$ queueing system with a preemptive resume, preemptive repeat and a preemptive partial repeat priority discipline. These authors also provide expressions for the generating functions of idle and busy periods enabling study of preemptive priority systems with more than two classes. Some authors consider a mixing of different disciplines. For instance, Kim and Chae [6] study a priority queue where service can only be preempted if the elapsed part of the service does not exceed a certain duration threshold.

As already mentioned, interruptions can also result from server failures or breakdowns. Some of the authors of the interruption models for priority queues discussed above exemplify that their models can also be applied when interruptions are triggered by server breakdowns instead of by high-priority customers. Evidently, server breakdowns have also been studied outside of the priority queueing context. Notice that, for the sake of uniformity, we hold on to the priority queueing terminology to indicate how service is reengaged after interruptions. Federgruen and Green [7] provide bounds and approximations for the $M/G/1$ queue with generally distributed on- and off-times and a preemptive resume discipline. Generally distributed on- and off-periods were also considered by Bruneel [8] for discrete-time queueing systems but with single slot service times such that there is no service preemption. Lee [9] investigates a similar system but with a Markovian interruption process. Núñez Queija [10] considers a processor sharing queue with Poisson breakdowns and preemptive resume. More recently, Balcioğlu et al. [11] approximate a $GI/D/1$ queue with correlated server breakdowns and preemptive resume by studying a similar system with an interruption process with (independent) hyper-exponential on-times and general off-times. Fiems et al. [12] study the $M/G/1$ queue where the server is both subjected to preemptive resume breakdowns and either preemptive repeat different or preemptive repeat identical breakdowns. Multiple server queues with Poisson arrival and breakdown process and exponential service times are studied by Mitrany and Avi-Itzhak [13] and Neuts and Lucantoni [14]. In the former contribution, server repair starts immediately and repair times are exponentially distributed, while in the latter contribution servers are repaired only when a number of servers have broken down.

The remainder of this contribution is organized as follows. The queueing model is described in detail in the next section. In section 3, we exploit the independence of arrival and interruption processes to simplify the analysis. When the interruption process is independent of the arrival and service processes, a queueing problem with server interruptions can be broken down into two separate problems: determination of the impact of the interruptions on customer service times and the analysis of the queueing system without interruptions. Next, some numerical examples are discussed illustrating the operation of the queueing system. Conclusions are drawn in the final section.

2 Model

We consider a discrete-time queueing system with a single server subject to interruptions. The queue is assumed to have infinite capacity. Time is divided into fixed-length intervals or slots. Arriving customers are stored in the queue. Service of customers is synchronized at slot boundaries. Consequently, customers can only start service at a slot boundary and leave the system, at a slot boundary, one or more slots later. When we observe the system at a slot, this is after the possible departure at the previous slot boundary but before any arrivals.

The number of arrivals at the system at consecutive slots are modelled by an independent and identically-distributed (i.i.d.) sequence of non-negative random variables A_k. The probability mass function (pmf) a_n denotes the probability that A_k takes the value n and the corresponding probability generating function (pgf) is given by

$$A(z) = \sum_{n=0}^{\infty} \Pr[A_k = n] z^n = \sum_{n=0}^{\infty} a_n z^n \,. \tag{1}$$

Similarly, the number of slots required by consecutive service times are characterized by the sequence of i.i.d. positive random variables S_k with pmf s_n and pgf $S(z)$.

The server is not permanently available for customers. After a slot where the server was available, it remains available with probability α, or, with probability $1 - \alpha$, it starts a vacation period of n slots according to the pmf b_n with corresponding pgf $B(z)$. The consecutive vacation periods are independent. For ease of notation, we also introduce the following pgf of a "server unavailability period",

$$N(z) = \alpha + (1 - \alpha) B(z) \,. \tag{2}$$

In the remainder, the server is said to be "free" when it is neither serving a customer nor unavailable. Notice that in the context of priority queues, this corresponds with the natural meaning of a free server: the server is neither serving customers of the class under consideration, nor serving customers with a higher priority.

If the server becomes unavailable (and thus leaves for a vacation) during an ongoing service, the elapsed part of this service time is lost and the service needs

to be repeated. The same service time is to be repeated with probability $1-\gamma$ or, with probability γ, a new service sample is drawn. Let $S_{k,i}$ denote the ith service attempt of the kth customer and let $\hat{S}_{k,i}$ denote a doubly indexed sequence of i.i.d. random variables distributed as S_k. For each k, the sequence of consecutive service attempts is a DAR(1) process, characterized by the equation

$$S_{k,i} = (1-\beta_{k,i})S_{k,i-1} + \beta_{k,i}\hat{S}_{k,i}\,, \tag{3}$$

where $\beta_{k,i}$ is a doubly indexed sequence of Bernoulli random variables with $\Pr[\beta_{k,i}=1]=\gamma$ and $\Pr[\beta_{k,i}=0]=1-\gamma$ and evidently $S_{k,1}=S_k$. Informally, this can be written as

$$S_{k,i} = \begin{cases} S_{k,i-1} & \text{with probability } 1-\gamma\,, \\ \hat{S}_{k,i} & \text{with probability } \gamma\,. \end{cases} \tag{4}$$

Note that this process is completely defined by the pgf $S(z)$ (and thus the pmf s_n) of the service times and the probability γ that the next service time is a new sample.

Invoking the moment-generating property of pgfs produces information about the distribution. Let the mean and variance of a generic random variable X, with pgf $X(z)$, be denoted by μ_X and σ_X^2 respectively. For instance, the mean and variance of the number of arriving customers per slot are respectively given by

$$\mu_A = \sum_{n=0}^{\infty} n a_n = A'(1)\,, \tag{5}$$

$$\sigma_A^2 = \sum_{n=0}^{\infty} a_n (n-\mu_A)^2 = A''(1) + A'(1) - A'(1)^2\,. \tag{6}$$

Here $X'(1)$ denotes the derivative of $X(z)$ with respect to z, evaluated in $z=1$.

Analogously, all moments of the random variables in this paper can be calculated from their pgf. In this manner, μ_S, σ_S^2, μ_B, σ_B^2, μ_N and σ_N^2 represent the mean and variance of the length of the customer service times, of the server vacation period and of the server unavailability period respectively. For further use, we introduce the symbol ν for the relative amount of available slots,

$$\nu = \frac{1}{1+(1-\alpha)\mu_B} = \frac{1}{1+\mu_N}\,. \tag{7}$$

3 Analysis

First, the interruption process is studied and the service completion time of a random customer is obtained. Next, the queueing analysis is performed and the system content and delay are subsequently determined.

3.1 Service completion time

Consider the kth customer and let his service completion time be defined as the number of slots between the start of the slot where he receives service for the first time and the end of the slot where he leaves the queue. This evidently encapsulates all consecutive service attempts of this customer and any possible server vacations between these attempts. Let $T_{k,i}\,(i \geq 0)$ denote the remaining service completion time of the kth customer after the ith interruption period. The entire service completion time of the kth customer is evidently equal to $T_{k,0}$. Furthermore, let $G_{k,i}\,(i \geq 1)$ denote the length of the ith available period during this service completion time. Moreover, let $B_{k,i}\,(i \geq 1)$ denote the length of the ith interruption period during this service completion time. Note that a (remaining) service completion time always starts with an available period as a service attempt starts when the server is free (and thus available). We establish,

$$T_{k,i} = \begin{cases} S_{k,i+1} & G_{k,i+1} \geq S_{k,i+1} \\ G_{k,i+1} + B_{k,i+1} + T_{k,i+1} & G_{k,i+1} < S_{k,i+1} \end{cases}, \tag{8}$$

as service is interrupted if its length exceeds the available period.

Let $T(z|n)$ denote the pgf of $T_{k,i}$ given that $S_{k,i+1} = n$ and let $T(z)$ denote the unconditional pgf of $T_{k,0}$. Notice that the distribution (and therefore also the pgf) of $T_{k,i}$ given $S_{k,i}$ does not depend on k and i. We have,

$$T(z|n) = \alpha^{n-1} z^n + \sum_{j=1}^{n-1} \alpha^{j-1}(1-\alpha) z^j B(z)(\gamma\, T(z) + (1-\gamma) T(z|n))\,. \tag{9}$$

The server is available at the start of this period so the service of n slots is completed if the server remains available for another $n-1$ slots. If this service is interrupted after $j\,(j \leq n)$ slots, the service completion time is augmented with a server vacation period and a next attempt at serving the customer is taken. For this next attempt, the required service time for this customer remains the same with probability $1-\gamma$ or a new service sample is drawn (with probability γ). When service is resampled, $T_{k,i}$ has the same pgf as $T_{k,0}$, namely $T(z)$.

From (9), some simple math produces

$$T(z|n) = \frac{\alpha^{n-1} z^n (1-\alpha z) + (1-\alpha)(1-(\alpha z)^{n-1}) z B(z) \gamma\, T(z)}{1 - \alpha z - (1-\gamma)(1-\alpha)(1-(\alpha z)^{n-1}) B(z) z}\,. \tag{10}$$

By summing over the service times with respect to the service time distribution, we find,

$$T(z) = \sum_{n=1}^{\infty} s_n \, \frac{\alpha^{n-1} z^n (1-\alpha z) + (1-\alpha)(1-(\alpha z)^{n-1}) z B(z) \gamma\, T(z)}{1 - \alpha z - (1-\gamma)(1-\alpha)(1-(\alpha z)^{n-1}) B(z) z}\,. \tag{11}$$

Finally, solving for $T(z)$ yields,

$$T(z) = \frac{T_n(z)}{1 - T_d(z)}\,, \tag{12}$$

with,

$$T_n(z) = \sum_{n=1}^{\infty} s_n \frac{\alpha^{n-1} z^n (1-\alpha z)}{1-\alpha z-(1-\gamma)(1-\alpha)(1-(\alpha z)^{n-1})B(z)z}\,,$$
$$T_d(z) = \sum_{n=1}^{\infty} s_n \frac{(1-\alpha)(1-(\alpha z)^{n-1})zB(z)\gamma}{1-\alpha z-(1-\gamma)(1-\alpha)(1-(\alpha z)^{n-1})B(z)z}\,. \tag{13}$$

Unfortunately, this expression is not explicit due to the presence of the infinite sums.

The expression corresponds to the service completion time in the PRI operation mode for $\gamma = 0$ and to the PRD operation mode for $\gamma = 1$. This yields

$$T(z)_{PRI} = \sum_{n=1}^{\infty} s_n \frac{\alpha^{n-1} z^n (1-\alpha z)}{1-\alpha z-1-\alpha)(1-(\alpha z)^{n-1})B(z)z}\,, \tag{14}$$

$$T(z)_{PRD} = \frac{S(\alpha z)(1-\alpha z)}{\alpha(1-\alpha z)-(1-\alpha)B(z)\big(\alpha z - S(\alpha z)\big)}\,. \tag{15}$$

In [15, Eq. 2.184, Eq. 2.188] and [5, Eq. 10], the effective service time, the sum of the service completion time and a server unavailability period, is computed for PRI and PRD. The expressions above can thus be verified, as multiplying them by $N(z)$ yields the effective service time. Also, note that the expression for PRD is explicit.

Recall that the moment-generating property of pgfs produces

$$\mu_T = T'(1) = \frac{T_n'(1) + T_d'(1)}{T_n(1)}\,, \tag{16}$$

where we used that, as $T(z)$ is a pgf, $T(1) = 1$ implies $T_n(1) + T_d(1) = 1$. By truncating the infinite sums appearing in $T_n(1)$, $T_n'(1)$ and $T_d'(1)$ at i, the smallest positive integer where $(\mu_T \restriction^i) - (\mu_T \restriction^{i-1}) < 10^j$, μ_T can be approximated with arbitrary precision (in function of j). Here, $(\mu_T \restriction^i)$ represents that for computing μ_T all infinite sums were truncated at i. Analogously, higher moments of the service completion time can be approximated.

3.2 Queue Content

First, the queue content at departure instants is calculated. Let $U_{d,k}$ and $U_{n,k}$ respectively denote the queue content at the kth departure instant and at the first slot the server is available following the kth departure and let $U_{d,k}(z)$ and $U_{n,k}(z)$ denote the corresponding pgfs. This yields,

$$U_{n,k}(z) = U_{d,k}(z)N(A(z))\,. \tag{17}$$

Note that $U_{d,k}$ and $U_{n,k}$ coincide with probability α.

Consider the first slot the server is available following the departure of customer k. Customer $k+1$ starts service at this slot, if the queue is not empty.

However, if the queue is empty, this (available) slot is followed by a possible server vacation and then an available slot. Service of the $k+1$th customer starts in this slot if the queue is not-empty, this is if packets have arrived during a slot followed by a server unavailability period. If the queue is still empty, this process is repeated until packets arrive in the period between two consecutive available slots.

These observations yield

$$U_{d,k+1}(z) = \big(U_{n,k}(z) - U_{n,k}(0)\big)\frac{T(A(z))}{z} + U_{n,k}(0)\frac{A(z)N(A(z)) - A(0)N(A(0))}{1 - A(0)N(A(0))}\frac{T(A(z))}{z}\,. \quad (18)$$

In view of equations (17) and (18), one sees that $U_{d,k}$ satisfies a Lindley-type stochastic recursion. By means of a Loynes-type argument, it is then easy to establish that there exist an almost surely finite steady-state solution if the time required to process a customer exceeds the customer inter-arrival time. This is, if the load $\rho = \mu_A(\mu_T + \mu_N) < 1$.

Let $U_d(z)$ denote the pgf of the queue content at departure epochs in steady state. Substituting (17) in (18) provides

$$U_d(z) = \frac{U_d(0)N(A(0))\big(1 - A(z)N(A(z))\big)T(A(z))}{\big(1 - A(0)N(A(0))\big)\big(T(A(z))N(A(z)) - z\big)}\,. \quad (19)$$

Normalization ($U_d(1) = 1$) produces

$$U_d(0) = \frac{\nu(1-\rho)\Big(1 - A(0)N(A(0))\Big)}{\mu_A N(0)}\,. \quad (20)$$

Substituting (20) in (19) yields

$$U_d(z) = \frac{\nu(1-\rho)\big(A(z)N(A(z)) - 1\big)T(A(z))}{\mu_A\big(z - T(A(z))N(A(z))\big)}\,. \quad (21)$$

We now determine the pgf of the queue content at random slots. Let $U_r(z)$ and $U_a(z)$ denote the pgfs of the queue content at random slots and arrival instants respectively. In [16], it is established that the queue content at the arrival of a certain customer is the sum of the queue content at the beginning of his arrival slot (which is equivalent to a random slot due to the independence of the arrivals from slot to slot) and the customers arriving in the same slot as but before the considered customer. Let $\hat{A}(z)$ denote the pgf of the number of customers arriving in the same slot as but before a certain customer. The observation above yields

$$U_r(z) = \frac{U_a(z)}{\hat{A}(z)} = U_a(z)\frac{\mu_A(z-1)}{A(z)-1}\,. \quad (22)$$

Furthermore, Burke's Theorem [17] states that the queue content at arrival and departure instants are statistically indistinguishable. This is

$$U_d(z) = U_a(z)\,. \tag{23}$$

Combining these two well-known results with (21) enables the determination of the queue content at random slots as

$$U_r(z) = \frac{\nu(1-\rho)(z-1)\big(A(z)N(A(z))-1\big)T(A(z))}{\big(A(z)-1\big)\big(z-T(A(z))N(A(z))\big)}\,. \tag{24}$$

Note that the stability condition $\rho < 1$ corresponds to $U_r(0) > 0$. Finally, the mean queue content at random slots is given by

$$\begin{aligned}\mu_{U_r} = \frac{\rho}{2} &+ \frac{\nu}{2}\mu_A(1-\alpha)\big(\mu_B^2(2\alpha-1)+\sigma_B^2-\mu_B\big)\\ &+ \frac{\mu_A^2\big(\sigma_T^2+(1-\alpha)\sigma_B^2+\alpha(1-\alpha)\mu_B^2\big)+\sigma_A^2(\mu_T+(1-\alpha)\mu_B)}{2(1-\rho)}\,.\end{aligned} \tag{25}$$

3.3 Delay

Customer delay is defined as the number of slots between the end of the arrival slot of a customer and the end of the slot where that customer leaves the queue. Rather than directly calculating the delay of a single customer, we apply a method from [18], where the batch delay is calculated first. The batch delay is defined for all slots where there is at least one arrival, say an arrival slot. The batch delay starts at the end of an arrival slot and ends when the last customer of the batch arriving during that arrival slot leaves the system. Hence, the batch delay is the delay of a "batch customer" in a queueing system where all customer arrivals in a single slot are grouped to form a batch customer.

The pgf of the number of batch-customer arrivals per slot $A^*(z)$ is then given by

$$A^*(z) = A(0) + \big(1-A(0)\big)z\,. \tag{26}$$

Moreover, let the service completion time of a batch-customer $T^*(z)$ be given by

$$T^*(z) = \frac{A(T(z)N(z)) - A(0)}{N(z)\big(1-A(0)\big)}\,. \tag{27}$$

This pgf corresponds to the sum of the successive service completion times of all customers arriving in a slot with at least one arriving customer in the original system, supplemented by the (possible) server unavailability between these service completion times. Notice that the construction of the batch service completion times obeys the interruption process of the original queueing system. We now substitute $A(z) = A^*(z)$ and $T(z) = T^*(z)$ into equation (21) and let $U_d^*(z)$ denote the resulting pgf,

$$U_d^*(z) = \frac{\nu(1-\rho)\big(A^*(z)N(A^*(z))-1\big)T^*(A^*(z))}{\mu_{A^*}\big(z-T^*(A^*(z))N(A^*(z))\big)}\,. \tag{28}$$

By construction, $U_d^*(z)$ is the pgf of the number of batch-customers in the queue upon departure of such a batch customer.

Now, consider a certain (batch-)customer. All customers in the queue at his arrival instant leave the system before the customer himself (as customers are served in order of arrival). Hence, all customers in the queue at the departure of the considered customer have arrived during the delay of the considered customer Therefore, the batch-customer delay, with a pgf denoted by $D^*(z)$, is related to the queue content at departure instants by

$$U_d^*(z) = D^*(A^*(z))\,, \quad \text{or} \quad D^*(z) = U_d^*(\frac{z - A(0)}{1 - A(0)})\,, \tag{29}$$

by using the definition of $A^*(z)$.

Finally, we can relate the delay of a random customer to the delay of the batch to which it belongs by taking into account the position of the customer within its batch. Observe the delay of a customer in the original system and the delay of the corresponding batch-customer (of which the customer is a part) in the alternative system. Instead of the service completion time of the entire batch-customer, only the part of the batch before the considered customer contributes to the delay of this customer. Therefore, the pgf of the customer delay in the original system is given by

$$\begin{aligned} D(z) &= \frac{D^*(z)}{T^*(z)} \hat{A}(T(z)N(z))T(z) \\ &= \frac{U_d^*(\frac{z-A(0)}{1-A(0)})}{T^*(z)} \hat{A}(T(z)N(z))T(z) \\ &= \frac{\nu}{\mu_A}(1-\rho)(1-zN(z))\frac{T(z)}{T(z)N(z)-1}\frac{A(T(z)N(z))-1}{A(T(z)N(z))-z}\,. \end{aligned} \tag{30}$$

By the moment-generating property of probability generating functions, moments of the customer delay can be calculated. In particular, the mean customer delay is given by

$$\begin{aligned} \mu_D = \frac{\rho}{2\mu_A} + \frac{\nu}{2}(1-\alpha)\big(\mu_B^2(2\alpha-1) + \sigma_B^2 - \mu_B\big) \\ + \frac{\mu_A^2\big(\sigma_T^2 + (1-\alpha)\sigma_B^2 + \alpha(1-\alpha)\mu_B^2\big) + \sigma_A^2(\mu_T + (1-\alpha)\mu_B)}{2\mu_A(1-\rho)}\,. \end{aligned} \tag{31}$$

Note that Little's theorem [19] holds as $\mu_{U_r} = \mu_A \mu_D$.

4 Numerical Examples

This section performs a quantitative analysis of some interesting system parameters. Let the number of arriving customers in a slot occur according to a Poisson

process and assume that the duration of a server vacation is geometrically distributed with parameter β. Consequently,

$$A(z) = e^{\mu_A(z-1)}\,, \tag{32}$$

$$B(z) = \frac{(1-\beta)z}{1-\beta z}\,. \tag{33}$$

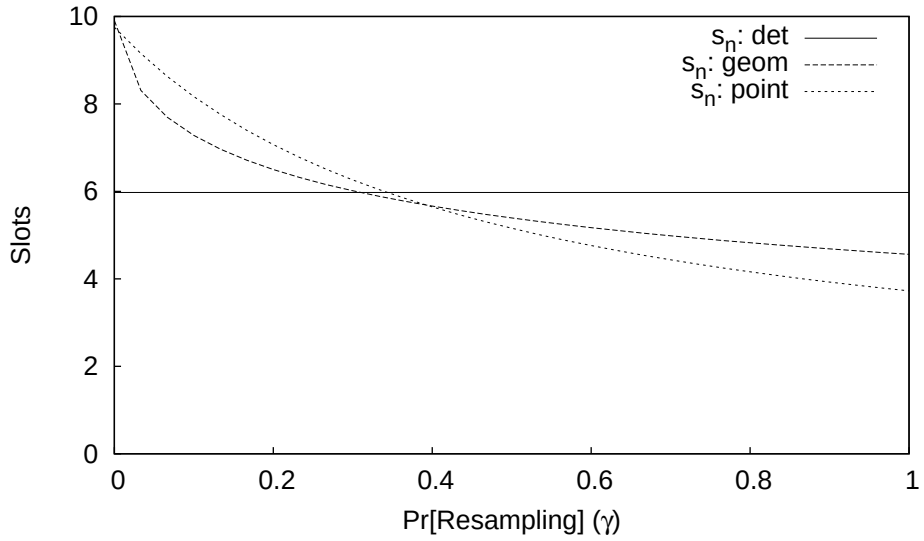

Fig. 1. Average service completion time versus resampling probability for various service time distributions

First, the effect of the resampling probability γ and the pmf of the service times s_n on the average service completion time is investigated. We consider three different distributions for the service times, all with average service time 4 slots but with different amounts of variance. We have the deterministic distribution, the geometric distribution and a distribution with all mass in two points. The pmf of the service time is then respectively given by

$$\text{deterministic: } s_n = \begin{cases} 1 & n = 4 \\ 0 & n \neq 4 \end{cases}\,, \tag{34}$$

$$\text{geometric: } s_n = 1/4(1-1/4)^{n-1}\,, \quad n \geq 1\,, \tag{35}$$

$$\text{mass-point: } s_n = \begin{cases} 2/3 & n = 1 \\ 1/3 & n = 10 \\ 0 & \text{otherwise} \end{cases}\,. \tag{36}$$

Consider the following system parameters: $\mu_A = 0.1$, $\alpha = 0.85$, $\beta = 0.2$. In figure 1, the average service completion time is plotted in function of the resampling probability γ for the three different service times. Evidently, resampling has no effect when the service times are deterministic. Resampling has a considerable impact, even for smaller values of γ. Therefore, queueing systems with even a small probability of service resampling cannot be approximated accurately by preemptive repeat identical. Also note that the higher the variance of the service times, the greater the effect of resampling. This is due to the fact that very long service times are almost always resampled into shorter service times. Consequently, the customer will leave the system earlier and this effect evidently increases with the resampling probability γ.

In the remainder, we will use geometrically distributed service times with parameter δ. Thus,

$$s_n = \delta(1-\delta)^{n-1}\,, \quad n \geq 1\,. \tag{37}$$

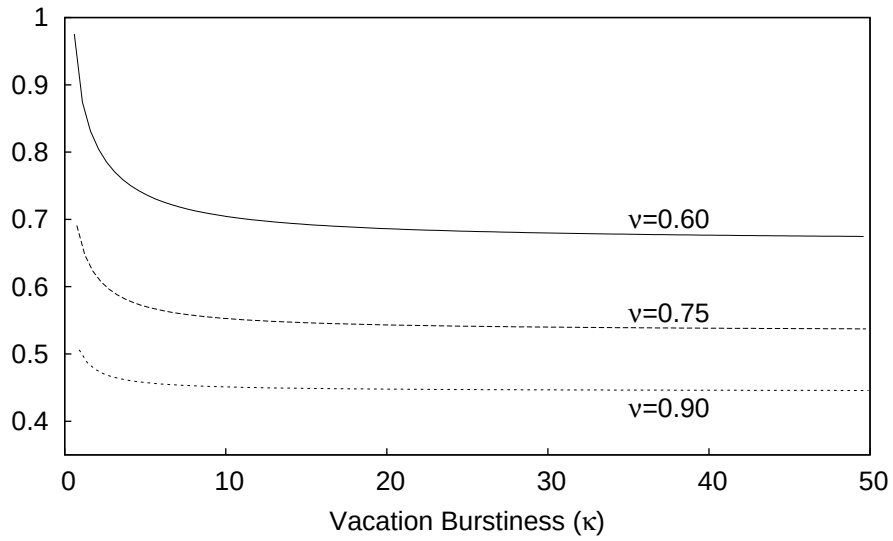

Fig. 2. Load ρ versus vacation burstiness κ

Finally, the effect of the vacation process on system performance is investigated. When characterizing this process, it is often more convenient to use ν, the fraction of available slots, and κ, the vacation burstiness, instead of α and

β. They are related by

$$\nu = \frac{1-\beta}{2-\alpha-\beta}\,, \tag{38}$$

$$\kappa = \frac{1}{2-\alpha-\beta}\,. \tag{39}$$

Note that by definition $\max(\nu, 1-\nu) \leq \kappa \leq \infty$ and that fixing ν and κ fixes α and β and vice versa.

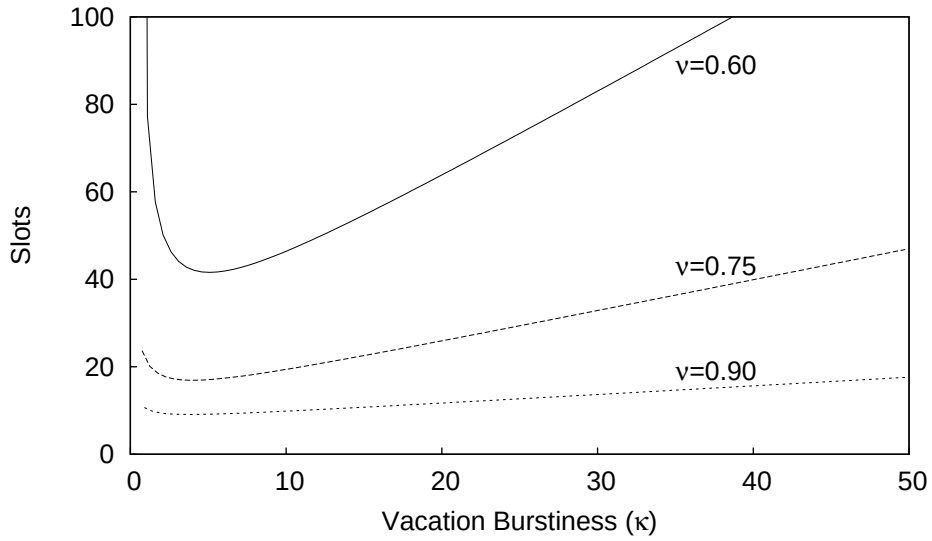

Fig. 3. Average customer delay μ_D versus vacation burstiness κ

Consider the following system parameters: $\mu_A = 0.1$, $\delta = 0.25$, $\gamma = 0.5$. The average customer delay μ_D (figure 3) and the (corresponding) load ρ (figure 2) are plotted versus the vacation burstiness κ for three different values of ν. For a fixed value of ν, smaller values of κ correspond to more yet shorter vacations while larger values of κ induce less but longer vacations. This explains the decreasing load ρ because service is interrupted less frequently as κ increases and hence the number of vacations decreases. For small values of κ, the average delay μ_D exhibits similar behavior. However, another effect takes over as κ increases: the delay increases as the vacations become more bursty and lengthy vacations elongate the delays of all packets in the queue. In contrast to the average delay, the load does not exhibit this behavior because it is only dependent on the mean values of the interruption process (μ_T and thus μ_B) and the mean service completion times decrease with κ. The average delay on the other hand is also

affected by the corresponding variances. Furthermore, these figures exemplify that a larger fraction of available slots ν (evidently) yields a smaller load ρ and shorter average delay μ_D.

5 Conclusions

We have proposed a hybrid preemptive repeat interruption discipline that encapsulates both the preemptive repeat identical and the preemptive repeat different disciplines. Subsequently, a discrete-time queueing system with such server interruptions was studied. By the introduction and analysis of so-called service completion times, the complexity of the analysis was reduced. Our results include a.o. the probability generating functions and moments of queue content and delay. Finally, by means of some numerical examples, the influence of the interruption discipline on system performance measures was investigated and we can conclude that in most situations even a small amount of resampling has considerable impact on system performance.

References

1. White, H., Christie, L.: Queuing with preemptive priorities or with breakdown. Operations Research **6**(1) (1958) 79–95
2. Avi-Itzhak, B., Naor, P.: Some queuing problems with the service station subject to breakdown. Operations Research **11**(3) (1963) 303–319
3. Thiruvengadam, K.: Queuing with breakdowns. Operations Research **11**(1) (1963) 62–71
4. Gaver Jr., D.: A waiting line with interrupted service, including priorities. Journal of the Royal Statistical Society **B24** (1962) 73–90
5. Fiems, D., Steyaert, B., Bruneel, H.: Discrete-time queues with generally distributed service times and renewal-type server interruptions. Performance Evaluation **55**(3-4) (2004) 277–298
6. Kim, K., Chae, K.: Discrete-time queues with discretionary priorities. European Journal of Operational Research **200**(2) (2010) 473–485
7. Federgruen, A., Green, L.: Queueing systems with service interruptions. Operations Research **34**(5) (1986) 752–768
8. Bruneel, H.: A general treatment of discrete-time buffers with one randomly interrupted output line. European Journal of Operational Research **27**(1) (1986) 67–81
9. Lee, D.: Analysis of a single server queue with semi-Markovian service interruption. Queueing Systems **27**(1–2) (1997) 153–178
10. Núñez Queija, R.: Sojourn times in a processor sharing queue with service interruptions. Queueing Systems **34**(1–4) (2000) 351–386
11. Balcioğlu, B., Jagerman, D.L., Altiok, T.: Approximate mean waiting time in a GI/D/1 queue with autocorrelated times to failures. IIE Transactions **39**(10) (October 2007) 985–996
12. Fiems, D., Maertens, T., Bruneel, H.: Queueing systems with different types of interruptions. European Journal of Operations Research **188** (2008) 838–845

13. Mitrany, I., Avi-Itzhak, B.: A many-server queue with service interruptions. Operations Research **16** (1968) 628–638
14. Neuts, M., Lucantoni, D.: Markovian queue with N-servers subject to breakdowns and repairs. Management Science **25**(9) (1979) 849–861
15. Fiems, D.: Analysis of discrete-time queueing systems with vacations. PhD thesis, Ghent University (2004)
16. Bruneel, H.: Performance of discrete-time queuing-systems. Computers & Operations Research **20**(3) (1993) 303–320
17. Takagi, H.: Queueing Analysis; A foundation of performance evaluation, volume 1: Vacation and priority systems, part 1. Elsevier Science Publishers (1991)
18. Takagi, H.: Queueing Analysis; A foundation of performance evaluation, volume 3: Discrete-time systems. Elsevier Science Publishers, Amsterdam (1993)
19. Fiems, D., Bruneel, H.: A note on the discretization of Little's result. Operations Research Letters **30**(1) (2002) 17–18