System occupancy of a two-class batch-service queue with class-dependent variable server capacity

Jens Baetens[1], Bart Steyaert[1], Dieter Claeys[1,2], and Herwig Bruneel[1]

[1]SMACS Research Group, Dept. of Telecommunications and Information Processing,
Ghent University, Ghent, Belgium
[2]Dept. of Industrial Systems Engineering and Product Design
Ghent University, Zwijnaarde, Belgium

Abstract. Due to their wide area of applications, queueing models with batch service, where the server can process several customers simultaneously, have been studied frequently. An important characteristic of such batch-service systems is the size of a batch, that is the number of customers that are processed simultaneously. In this paper, we analyse a two-class batch-service queueing model with variable server capacity, where all customers are accommodated in a common first-come-first served single-server queue. The server can only process customers that belong to the same class, so that the size of a batch is determined by the number of consecutive same-class customers. After establishing the system equations that govern the system behaviour, we deduce an expression for the steady-state probability generating function of the system occupancy at random slot boundaries. Also, some numerical examples are given that provide further insight in the impact of the different parameters on the system performance.

Keywords: Discrete Time · Batch Service · Two Classes · Variable Server Capacity · Queueing

1 Introduction

In telecommunication applications, a single server can often process multiple customers (i.e. data packets) simultaneously in a single batch. An important characteristic of such batch-service systems is the maximum size of a batch, that is the maximum number of customers processed simultaneously. In many batch-service systems this number is assumed to be a constant [1–5]. However, in practice, the maximum batch size or capacity of the server can be variable and stochastic, a feature that has been incorporated in only a few papers. Chaudhry and Chang analysed the system content at various epochs in the $Geo/G^Y/1/N+B$ model in discrete time, where Y denotes the stochastic capacity of the server, which is upper-bounded by B, and N is the maximum queue capacity [6]. Furthermore, Pradhan et al. obtained closed-form expressions for the queue length distribution at departure epochs for the discrete-time $M/G_r^Y/1$ queue where the service process depends on the batch size [7]. A similar feature in the models from Chaudhry and Chang, and Pradhan et al. is that the capacity of a batch is independent of the queue length and of the capacities of the previous batches. On the contrary, Germs and van Foreest have recently developed an algorithmic method for the performance evaluation of the continuous-time $M(n)^{X(n)}/G(n)^{Y(n)}/1/K+B$ queue [8]. In that model, both the arrival rate and service process (the service times as well as the capacities) depend on the queue size.

Another feature of the above models is that customers are indistinguishable, i.e., they all are of the same type. Although in many types of queueing systems several customer classes are included to account for customer differentiation, only a few papers on batch service consider multiple customer classes. Reddy et al. study a multi-class batch-service queueing system with Poisson arrivals and a priority scheduling discipline, in the context of an industrial repair shop where the most critical machines are repaired first [9]. Boxma et al. study a polling system with Poisson arrivals and batch service [10]. In this case, each customer class has a dedicated queue and the server visits the different queues in a cyclic manner. Boxma et al. focus on the influence of a number of different gating policies on the performance. Dorsman et al. study a polling system with a renewal arrival process and batch service, where the batches are created by accumulation stations before they are added to a queue [11]. Such a system can be used when a single server processes multiple product types with batching constraints. Dorsman et al. focus on optimizing the batch sizes of each class.

In this paper, we analyse a two-class discrete-time batch-service queueing model, with a variable service capacity that depends on the queue size and on the specific classes of the successive customers. To the best of our knowledge, the combination of batch service with variable capacity and multiple customer classes has not appeared in the literature before. Whereas in the mentioned papers about priority queueing and polling systems the customers of different classes are accomodated in different queues, the customers of all classes are accommodated in a common queue here. When the server becomes available, it will simultaneously process the customer at the head of the queue, and all successive customers that are of the same class as the head customer. This, for instance, means that if the first customer is of class A, all of the following class A customers are also grouped in the batch that will be taken into service, until the next customer is of class B. Applications of this server can be found in manufacturing environments or telecommunication systems, where customers with the same system parameter, such as the required temperature or the destination of the customer, can be processed simultaneously on a FCFS-basis.

The paper is structured as follows. In Section 2 we describe the discrete-time two-class queueing model with batch service in detail. This system consists of a single First-Come-First-Served (FCFS) queue of infinite size, and a single batch server with a variable capacity. In Section 3 we establish the system equations, from which we deduce the stability condition, and derive a closed-form expression for the steady-state probability generating function (pgf) of the system occupancy at random slot boundaries. Next, using the expressions obtained in Section 3, we evaluate the behaviour of the system through some numerical examples in Section 4. Our conclusions are presented in Section 5.

2 Model description

Let us consider a discrete-time two-class queueing system with infinite queue size, and a batch server whose capacity is stochastic. The classes of the customers are denominated as A and B. Arriving customers are inserted at the tail of the queue. When the server is or becomes available and finds a non-empty queue, a new service is initiated. The size of the batch is then determined by the number of consecutive customers at the front of the system that are of the same class. More specifically, the server starts serving a batch of n customers if and only if one of the following two cases occurs:

- Exactly n customers are present and they are all of the same class.
- More than n customers are present, the n customers at the front of the queue are of the same class and the $(n+1)$-th customer is of the other class.

We define the class of a batch as the class of the customers within it.

The aggregated numbers of customer arrivals in consecutive slots are modelled as a sequence of independent and identically distributed (i.i.d.) random variables, with common probability mass function (pmf) $e(n)$ and pgf $E(z)$. The mean aggregated number of customer arrivals per slot is denoted as λ. A random customer is of class A with probability (w.p.) σ and of class B w. p. $1-\sigma$ regardless of the classes of other customers. The service time of a batch is always a single slot, independently of both the class of the batch and its size.

3 Analysis

In this section, we first determine the system equations that capture the system behaviour. Then we analyse the conditions for stability, and we establish the steady-state pgf of the system occupancy, that is the number of customers in the system at the beginning of a slot, including those in the batch that will be served during this slot (if any).

3.1 System Equations

In this subsection, we give the system equations that capture the behaviour of the system at successive slot boundaries. The number of customers in the system or the system occupancy at random slot boundaries is denoted by u_k. We also define the random variables $u_{I,k}$, $u_{A,k}$ and $u_{B,k}$ as the system occupancy at the boundary of slot k when the server respectively is idle or processes a class A or B batch.

If the server is idle during slot k, then the next slot is also an idle slot if there are no new arrivals. On the other hand, when there is at least one arrival, then the server will process a class A or B batch based on the class of the first arrival. This leads to the system equations if $u_{I,k} = 0$:

$$\begin{aligned} u_{I,k+1} &= 0, \text{ if } e_k = 0 \\ u_{A,k+1} &= e_k, \text{ if } e_k > 0 \;\&\; \text{first arrival of class A, w.p. } \sigma \\ u_{B,k+1} &= e_k, \text{ if } e_k > 0 \;\&\; \text{first arrival of class B, w.p. } 1-\sigma \;\;, \end{aligned} \tag{1}$$

where e_k is the number of customers that arrive during slot k, with pmf $e(n)$ and pgf $E(z)$.

On the other hand, if a class A batch is processed during a random slot k and there are $u_{A,k}$ customers in the system, then the system equations also depend on the size of the processed batch. In the first case, all waiting customers are served meaning that the batch size c_k is equal to $u_{A,k}$. This leads to a similar behaviour as for an idle slot. If the size of the batch is less than $u_{A,k}$ meaning that not all customers are processed simultaneously, then the customer at the head of the queue must be of the opposite class, implying that a class B batch is always processed during slot $k+1$. Summarized, we have

$$\begin{aligned} u_{I,k+1} =&0, \text{ if } e_k = 0 \;\&\; c_k = u_{A,k} \\ u_{A,k+1} =&e_k, \text{ if } e_k > 0 \;\&\; c_k = u_{A,k} \;\&\; \text{first arrival of class A (w.p. } \sigma) \\ u_{B,k+1} =&e_k, \text{ if } e_k > 0 \;\&\; c_k = u_{A,k} \;\&\; \text{first arrival of class B (w.p. } 1-\sigma) \\ u_{B,k+1} =&u_{A,k} - c_k + e_k, \text{ if } c_k < u_{A,k} \;\;, \end{aligned} \tag{2}$$

where $c_k > 0$ is the size of the batch being processed during slot k.

The case that a class B batch was processed during a random slot k leads to the counterpart system equations

$$\begin{aligned} u_{I,k+1} =&0, \text{ if } e_k = 0 \;\&\; c_k = u_{B,k} \\ u_{A,k+1} =&e_k, \text{ if } e_k > 0 \;\&\; c_k = u_{B,k} \;\&\; \text{first arrival of class A (w.p. } \sigma) \\ u_{B,k+1} =&e_k, \text{ if } e_k > 0 \;\&\; c_k = u_{B,k} \;\&\; \text{first arrival of class B (w.p. } 1-\sigma) \\ u_{A,k+1} =&u_{B,k} - c_k + e_k, \text{ if } c_k < u_{B,k} \;\;. \end{aligned} \tag{3}$$

3.2 Stability Condition

In order to find the stability condition, we analyse the system under the condition that the queue is saturated. In such a system, the batch server is never idle and the size of the processed batches is not limited by a lack of customers and therefore geometrically distributed. Because the server processes all same-class customers at the head of the queue, the server will alternate between processing class A and B batches, which means we can limit ourselves to considering 2 consecutive slots. The system is stable when the mean number of customer arrivals during two consecutive slots, which is equal to 2λ, is less than the mean number of customers processed during the same slots. The mean number of processed customers during two consecutive slots is the sum of the mean batch size of a class A and B batch. The batch size follows a geometric distribution with parameter σ (class A) or $1-\sigma$ (class B) respectively. The stability condition is then given by

$$2\lambda < \frac{1}{1-\sigma} + \frac{1}{\sigma} \ .$$

If σ is either 0 or 1, then the stability condition is reduced to $\lambda < \infty$, i.e., the system is always stable. This is as expected, since in this case all customers are of the same class, which means that no matter how many customers arrive, the server will always aggregate all waiting customers in a single batch. Also, if σ is equal to 0.5 then the maximum tolerable arrival rate reaches a minimum value.

We can also define the load ρ of the system as the fraction of the average number of arrivals versus the maximum allowed arrival rate, which leads to

$$\rho := \frac{2\lambda}{\frac{1}{1-\sigma} + \frac{1}{\sigma}} = 2\lambda\sigma(1-\sigma) < 1 \ . \tag{4}$$

3.3 System occupancy

Assuming the stability condition is met, we can define the pmf of u_k, the system occupancy at random slot boundaries, as

$$u(i) := \lim_{k\to\infty} \Pr[u_k = i] \ ,$$

with corresponding pgf

$$U(z) := \sum_{i=0}^{\infty} u(i) z^i \ .$$

We can split the generating function of the system occupancy $U(z)$ in three parts based on the state of the server (idle, processing a class A batch or class B batch). This leads to

$$U(z) = u_I + U_A(z) + U_B(z) \ , \tag{5}$$

where we introduced the following definitions

$$u_I := \lim_{k\to\infty} \Pr[u_{I,k} = 0] \ ,$$
$$U_A(z) := \sum_{i=1}^{\infty} \lim_{k\to\infty} \Pr[u_{A,k} = i] z^i \ ,$$
$$U_B(z) := \sum_{i=1}^{\infty} \lim_{k\to\infty} \Pr[u_{B,k} = i] z^i \ .$$

The first term in the right-hand side of Eq. 5 corresponds to the probability that the server is idle during a random slot. This probability u_I is found by invoking the system equations in Section 3.1.

$$\begin{aligned}u_I =&u_I E(0) + \lim_{k\to\infty}\sum_{i=1}^{\infty}\Pr[u_{A,k}=i]E(0)\sigma^{i-1}\\&+\lim_{k\to\infty}\sum_{i=1}^{\infty}\Pr[u_{B,k}=i]E(0)(1-\sigma)^{i-1}\\=&u_I E(0)+E(0)\frac{U_A(\sigma)}{\sigma}+E(0)\frac{U_B(1-\sigma)}{1-\sigma}\quad ,\end{aligned}$$

leading to

$$u_I =\frac{E(0)}{(1-E(0))}\left(\frac{U_A(\sigma)}{\sigma}+\frac{U_B(1-\sigma)}{1-\sigma}\right)\quad . \tag{6}$$

Based on the state of the server during the previous slot, we can split the second term of Eq. 5 as

$$\begin{aligned}U_A(z) =&E[z^{u_{A,k+1}}] = E[z^{u_{A,k+1}}I_{\{u_{I,k}=0\}}] + E[z^{u_{A,k+1}}I_{\{u_{A,k}>0\}}]\\&+E[z^{u_{A,k+1}}I_{\{u_{B,k}>0\}}]\quad ,\end{aligned}\tag{7}$$

where $I_{\{C\}}$ are indicator functions which are equal to 1 if event C occurs and zero otherwise. The first part of this equation gives the partial generating function in case of the server being idle in the previous slot. Using Eq. 1 we can write this function as

$$E[z^{u_{A,k+1}}I_{\{u_{I,k}=0\}}] = \sigma E[z^{e_k}I_{\{u_{I,k}=0,e_k>0\}}] = \sigma u_I(E(z)-E(0))\quad . \tag{8}$$

Analogously we can write the second part, invoking the system equations in Eq. 2, as

$$\begin{aligned}E[z^{u_{A,k+1}}I_{\{u_{A,k}>0\}}] =&\sigma E[z^{e_k}I_{\{u_{A,k}>0,c_k=u_{A,k},e_k>0\}}]\\=&(E(z)-E(0))\lim_{k\to\infty}\sigma\sum_{i=1}^{\infty}\sigma^{i-1}\Pr[u_{A,k}=i]\\=&(E(z)-E(0))U_A(\sigma)\quad .\end{aligned}\tag{9}$$

The last part of $U_A(z)$ corresponds to the case that a class B batch is processed during the previous slot. Using Eq. 3, we obtain the following equation

$$\begin{aligned}&E[z^{u_{A,k+1}}I_{\{u_{B,k}>0\}}]\\&\quad=\sigma E[z^{e_k}I_{\{u_{B,k}>0,c_k=u_{B,k},e_k>0\}}] + E[z^{u_{B,k}-c_k+e_k}I_{\{u_{B,k}>1,c_k<u_{B,k}\}}]\quad .\end{aligned}$$

If the number of customers in the system is equal to i, then the probability that the size of the batch is equal to i is given by $(1-\sigma)^{i-1}$ since we know the class of the first customer and the next $i-1$ customers must be of class B. On the other hand, the probability that the size of the served batch is equal to $j < i$ is given by $\sigma(1-\sigma)^{j-1}$ because the $(j+1)$-th customer must be of class A. This leads to

$$\begin{aligned}
&E[z^{u_{A,k+1}} I_{\{u_{B,k}>0\}}] \\
&\quad = \sigma(E(z)-E(0)) \sum_{i=1}^{\infty} (1-\sigma)^{i-1} \lim_{k\to\infty} \Pr[u_{B,k}=i] \\
&\quad\quad + \sigma E(z) \sum_{i=2}^{\infty} \sum_{j=1}^{i-1} z^{i-j} (1-\sigma)^{j-1} \lim_{k\to\infty} \Pr[u_{B,k}=i] \\
&\quad = \sigma(E(z)-E(0)) \frac{U_B(1-\sigma)}{1-\sigma} \\
&\quad\quad + \sigma E(z) \sum_{i=2}^{\infty} \lim_{k\to\infty} \Pr[u_{B,k}=i] \frac{(1-\sigma)z^i - z(1-\sigma)^i}{(1-\sigma)(z-(1-\sigma))} \\
&\quad = \sigma(E(z)-E(0)) \frac{U_B(1-\sigma)}{1-\sigma} \\
&\quad\quad + \frac{\sigma E(z)}{(1-\sigma)(z-(1-\sigma))} \left((1-\sigma)U_B(z) - zU_B(1-\sigma) \right) . \qquad (10)
\end{aligned}$$

By combining Eqs. 8, 9 and 10, we obtain the partial pgf of the system occupancy in a slot where a class A batch is processed.

$$\begin{aligned}
U_A(z) = {} & \sigma(E(z)-E(0)) \left(u_I + \frac{U_A(\sigma)}{\sigma} + \frac{U_B(1-\sigma)}{1-\sigma} \right) \\
& + \frac{\sigma E(z)}{(1-\sigma)(z-(1-\sigma))} \left((1-\sigma)U_B(z) - zU_B(1-\sigma) \right) . \qquad (11)
\end{aligned}$$

An analogous analysis leads to the partial generating function $U_B(z)$.

$$\begin{aligned}
U_B(z) = {} & (1-\sigma)(E(z)-E(0)) \left((u_I + \frac{U_A(\sigma)}{\sigma} + \frac{U_B(1-\sigma)}{1-\sigma} \right) \\
& + \frac{(1-\sigma)E(z)}{\sigma(z-\sigma)} \left(\sigma U_A(z) - zU_A(\sigma) \right) . \qquad (12)
\end{aligned}$$

By substituting Eq. 12 in Eq. 11 we obtain for $U_A(z)$

$$\begin{aligned}
U_A(z) = {} & \sigma(E(z)-E(0)) \left(u_I + \frac{U_A(\sigma)}{\sigma} + \frac{U_B(1-\sigma)}{1-\sigma} \right) - \frac{\sigma z E(z) U_B(1-\sigma)}{(1-\sigma)(z-(1-\sigma))} \\
& + \frac{\sigma E(z)}{z-(1-\sigma)} \left((1-\sigma)(E(z)-E(0)) \left(u_I + \frac{U_A(\sigma)}{\sigma} + \frac{U_B(1-\sigma)}{1-\sigma} \right) \right. \\
& \left. + \frac{(1-\sigma)E(z)}{z-\sigma} U_A(z) - \frac{(1-\sigma)zE(z)U_A(\sigma)}{\sigma(z-\sigma)} \right) .
\end{aligned}$$

We now multiply by $(z-\sigma)(z-1+\sigma)$ and put all terms that contain $U_A(z)$ in the left-hand side of the equation. Also using Eq. 6 to substitute u_I leads to

$$\begin{aligned}
&U_A(z)\Big((z-\sigma)(z-(1-\sigma))-\sigma(1-\sigma)E(z)^2\Big)\\
&\quad=\sigma(z-\sigma)\Big(z-1+\sigma+(1-\sigma)E(z)\Big)\frac{E(z)-E(0)}{1-E(0)}\left(\frac{U_A(\sigma)}{\sigma}+\frac{U_B(1-\sigma)}{1-\sigma}\right)\\
&\quad-\frac{\sigma(z-\sigma)zE(z)U_B(1-\sigma)}{1-\sigma}-(1-\sigma)zE(z)^2U_A(\sigma)\ . \qquad (13)
\end{aligned}$$

The analogous expression for class B then satisfies

$$\begin{aligned}
&U_B(z)\Big((z-\sigma)(z-(1-\sigma))-\sigma(1-\sigma)E(z)^2\Big)\\
&\quad=(1-\sigma)(z-1+\sigma)\Big(z-\sigma+\sigma E(z)\Big)\frac{E(z)-E(0)}{1-E(0)}\left(\frac{U_A(\sigma)}{\sigma}+\frac{U_B(1-\sigma)}{1-\sigma}\right)\\
&\quad-\frac{(1-\sigma)(z-1+\sigma)zE(z)U_A(\sigma)}{\sigma}-\sigma zE(z)^2U_B(1-\sigma)\ . \qquad (14)
\end{aligned}$$

The sum of Eqs. 6, 13 and 14 lead to the pgf of the system occupancy at random slot boundaries. This generating function is equal to

$$\begin{aligned}
&U(z)\Big((z-\sigma)(z-(1-\sigma))-\sigma(1-\sigma)E(z)^2\Big)\\
&\quad=u_I\Big((z-\sigma)(z-(1-\sigma))-\sigma(1-\sigma)E(z)^2\Big)+\frac{E(z)-E(0)}{1-E(0)}\\
&\quad\cdot\Big((z-\sigma)(z-1+\sigma)+\sigma(1-\sigma)(2z-1)E(z)\Big)\left(\frac{U_A(\sigma)}{\sigma}+\frac{U_B(1-\sigma)}{1-\sigma}\right)\\
&\quad-(1-\sigma)zE(z)U_A(\sigma)\Big(E(z)+\frac{z-1+\sigma}{\sigma}\Big)\\
&\quad-\sigma zE(z)U_B(1-\sigma)\Big(E(z)+\frac{z-\sigma}{1-\sigma}\Big)\ . \qquad (15)
\end{aligned}$$

The two remaining unknowns $U_A(\sigma)$ and $U_B(1-\sigma)$ in the pgf $U(z)$ are yet to be determined. With the theorem of Rouché, we can easily prove that the denominator of $U(z)$ has two zeros inside or on the unit circle. Each zero of the denominator must also be a zero of the numerator since generating functions are analytical functions inside the complex unit disk and bounded for $|z|=1$. In Eq. 15 we can easily see that $z=1$ is a zero of the denominator. The other zero can be calculated numerically. The equations provided by the zeros constitute a set of two linear equations for two unknowns. We also note that $z=1$ leads to the normalisation condition. By evaluating Eq. 15 at $z=1$ and applying l'Hôpital's rule we obtain

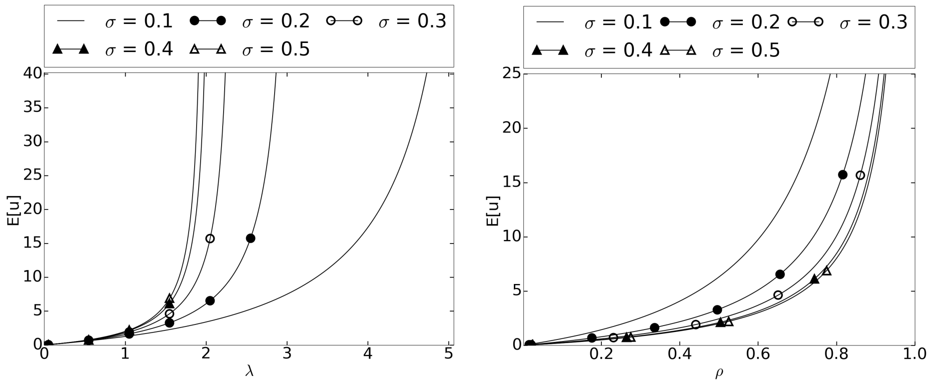
$$1=u_I+U_A(\sigma)\frac{1+\frac{2(1-\sigma)\lambda E(0)}{1-E(0)}}{1-2\sigma(1-\sigma)\lambda}+U_B(1-\sigma)\frac{1+\frac{2\sigma\lambda E(0)}{1-E(0)}}{1-2\sigma(1-\sigma)\lambda}\ .$$

4 Numerical results

In this section, we illustrate the results obtained in the previous section through numerical examples. In Fig. 1-3, we consider a geometric arrival process with mean arrival rate λ. The pgf $E(z)$ is equal to

$$E(z) = \frac{1}{1 + \lambda(1 - z)} \ .$$

The influence of the parameter σ on the average system occupancy $E[U]$ is shown in Fig. 1 as function of both the mean arrival rate and the load of the system. In Fig. 1a we note that smaller values of σ lead to a significant improvement of the performance of the system. This is caused by the inverse proportionality of the parameter σ to the size of the batches being processed. On the other hand, we observe in Fig. 1b that the mean system occupancy is larger for smaller values of σ when the server is operating under the same load, as defined in Eq. 4. This is the result of two conflicting effects. A smaller value of σ leads to a higher average number of customers that the server can process due to larger sequences of same-class customers, but also to a higher arrival rate to obtain the same load in the system. In Fig. 1b it is clear that the influence of the increased arrival rate is most significant, partially because the number of customers in the server are also part of the system occupancy and partially because there must be more customers waiting in the queue to create the larger batches that can be processed.



(a) Versus the mean arrival rate λ

(b) Versus the load ρ

Fig. 1: Influence of σ on the average system occupancy as a function of the arrival rate λ (a) and the load ρ (b).

A more detailed analysis of the influence of σ on the mean system occupancy when the server is operating under a certain load is shown in Fig. 2. We first observe that σ is symmetric around 0.5, which means that a probability of a class A customer being equal to σ or $1 - \sigma$ will lead to the same value for the mean system occupancy. If we look at $0.2 < \sigma < 0.8$ we see that the influence of σ is only significant for larger loads. This is because the performance of the server is limited by a lack of customers at lower loads for these values of σ. For σ closer to 0 or 1 we see that the average system occupancy is increases drastically, even for small loads. This is because the server can process larger batches and the arrival rate must increase to obtain the same load.

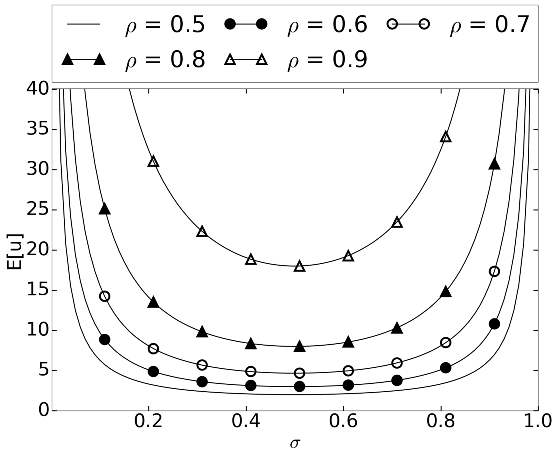


Fig. 2: Influence of σ on the mean system occupancy for a number of different loads

Another important characteristic of the system is the probability that a server is idle during a random slot. This probability is calculated according to Eq. 6 and depends on the probability that there are no arrivals during a slot and the probability that the server processes all customers during the same slot. In Fig. 3 we show this characteristic in terms of σ for a number of different loads. We observe that when σ approximates 0 or 1, that the server is almost never idle regardless of the load of the system. This occurs because the maximum allowed arrival rate is very large so that even small loads lead to a large mean arrival rate. A large mean arrival rate means that the probability that there are no arrivals is very small so that the server will almost always be able to start a service. We also observe that for σ closer to 0.5, the probability u_I is not very sensitive for variations of σ. This is caused by a conflict between the probability that there are no arrivals and the probability that all customers are processed. Values of σ closer to 0.5 means that the probability that there are no arrivals increases but the probability that all customers are processed decreases.

In a last example we examine the influence of the variance in the arrival process. Therefore, we consider an arrival process where the number of arrivals in an arbitrary slot is with probability α determined by a geometric distribution

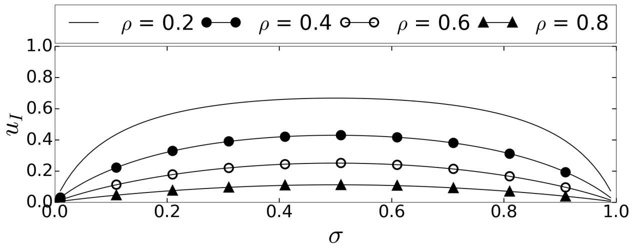


Fig. 3: Influence of σ on u_I, the probability that the server is idle, for the system under different loads

with mean $\frac{\lambda}{2\alpha}$ and with probability $1-\alpha$ by a geometric distribution with mean $\frac{\lambda}{2(1-\alpha)}$. The pgf that describes this arrival process is therefore given by

$$E(z) = \alpha \frac{1}{1+\frac{\lambda}{2\alpha}(1-z)} + (1-\alpha)\frac{1}{1+\frac{\lambda}{2(1-\alpha)}(1-z)} \quad .$$

The mean arrival rate of this arrival process is still equal to λ, and the variance of e, the number of arrivals during a random slot,

$$\mathrm{Var}[e] = \frac{\lambda^2}{2\alpha(1-\alpha)} + \lambda - \lambda^2 \quad .$$

This equation indicates that the variance is minimal for $\alpha = 0.5$, and approaches infinity for α close to 0 or 1. In Fig. 4, we plot the mean system occupancy as a function of α for values of σ as indicated and with a load of $\rho = 0.9$. We clearly observe the detrimental effect an increasing variance has on the mean system occupancy.

5 Conclusions

In this paper we have analysed a discrete-time two-class single-server queueing system with batch service. The size of the batches that are processed are determined by the number of customers in the queue and their respective classes. We have derived the steady-state pgf of the number of customers in the system at random slot boundaries. Using these results, we have demonstrated the impact of the various parameters of the arrival process on the average system occupancy and the probability that the server is idle.

There are a number of possible extensions that could be considered for this model. A first extension would be to find the probability generating function for the number of customers that are being processed by the batch server. In a second extension we could extend the model to use a class-dependent general service time distribution for class A and B batches. A further extension we mention is that we could introduce bursty behaviour of same class customers

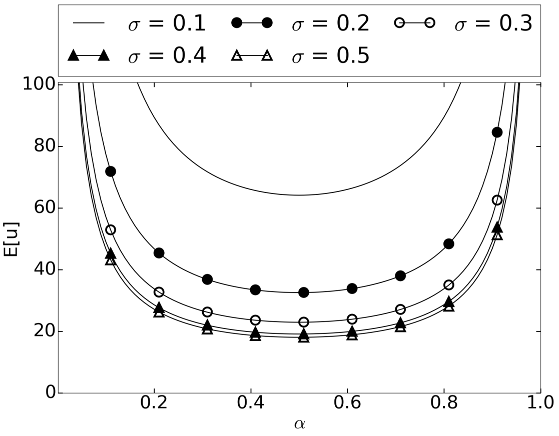


Fig. 4: Influence of α, which determines the variance in the arrival process, on the system occupancy for a number of values of σ and with a load of 0.9

by introducing correlation between the class of two consecutive customers. This can for instance be done by using a general 2-state Markov process to assign customer classes. This allows us to tweak the length of class A or B customers that arrive while maintaining a certain ratio of class A and B customers.

Acknowledgment. Dieter Claeys is a Postdoctoral Fellow with the Research Foundation Flanders (FWO-Vlaanderen), Belgium. Part of the research has been funded by the Interuniversity Attraction Poles Programma initiated by the Belgian Science Policy Office.

References

1. A. Banerjee and U.C. Gupta: Reducing congestion in bulk-service finite-buffer queueing system using batch-size-dependent service. Performance Evaluation, 69(1):53-70, 2012.
2. D. Claeys and B. Steyaert and J. Walraevens and K. Laevens and H. Bruneel: Tail distribution of the delay in a general batch-service queueing model. Computers and Operations Research,39:2733-2741, 2012.
3. D. Claeys and B. Steyaert and J. Walraevens and K. Laevens and H. Bruneel: Analysis of a versatile batch-service queueing model with correlation in the arrival process. Performance Evaluation, 70(4):300-316, 2013.
4. V. Goswami and J.R. Mohanty and S.K. Samanta: Discrete-time bulk-service queues with accessible and non-accessible batches. Applied Mathematics and Computation, 182:898-906, 2006.
5. W.W. Weng and R.C. Leachman: An improved methodology for real-time production decisions at batch-process work stations. Transactions on Semiconductor Manufacturing,6(3):219-225, 1993.

6. M.L. Chaudhry and S.H. Chang: Analysis of the discrete-time bulk-service queue $Geo/G^{Y}/1/N+B$. Operations Research Letters, 32(4):355-363, 2004.
7. Pradhan, S and Gupta, UC and Samanta, SK: Queue-length distribution of a batch service queue with random capacity and batch size dependent service: $M/G_{r}^{Y}/1$. OPSEARCH, pages 1-15, 2015.
8. R. Germs and N.D. Van Foreest: Analysis of finite-buffer state-dependent bulk queues. OR Spectrum, 35(3):563-583, 2013.
9. G.V.K. Reddy and R. Nadarajan and P.R. Kandasamy: A nonpreemptive priority multiserver queueing system with general bulk service and heterogeneous arrivals. Computers and operations research, 20(4):447-453, 1993.
10. O.J. Boxma and J. van der Wal and U. Yechiali: Polling with batch service. Stochastic Models, 24(4):604-625, 2008.
11. J.L. Dorsman and R.D. Van der Mei and E.M.M. Winands: Polling with batch service. OR Spectrum, 34:743-761, 2012.