STOCHASTIC DECOMPOSITION IN DISCRETE-TIME QUEUES
WITH GENERALIZED VACATIONS AND APPLICATIONS

Sofian De Clercq,
Wouter Rogiest, Bart Steyaert, Herwig Bruneel
SMACS Research Group, Ghent University
St.-Pietersnieuwstraat 41,
9000 Gent, BELGIUM

Abstract. For several specific queueing models with a vacation policy, the stationary system occupancy at the beginning of a random slot is distributed as the sum of two independent random variables. One of these variables is the stationary number of customers in an equivalent queueing system with no vacations. For models in continuous time with Poissonian arrivals, this result is well-known, and referred to as stochastic decomposition, with proof provided by Fuhrmann and Cooper. For models in discrete time, this result received less attention, with no proof available to date. In this paper, we first establish a proof of the decomposition result in discrete time. When compared to the proof in continuous time, conditions for the proof in discrete time are somewhat more general. Second, we explore four different examples: non-preemptive priority systems, slot-bound priority systems, polling systems, and fiber delay line (FDL) buffer systems. The first two examples are known results from literature that are given here as an illustration. The third is a new example, and the last one (FDL buffer systems) shows new results. It is shown that in some cases the queueing analysis can be considerably simplified using this decomposition property.

1. Introduction. Under certain conditions, the system content of a queueing system (i.e. the number of customers in it) with server vacations is distributed as the sum of two independent random variables, one of which is the stationary number of customers in an equivalent queueing system without vacations. This property, commonly denoted as stochastic decomposition, is both intriguing and elegant, since it harbors the possibility to solve for the system content distribution in such a vacation system much easier. Decomposition is intriguing since, although the conclusion is relatively simple, the underlying reasons are not. Moreover, the number of continuous-time queueing systems for which such decomposition results were obtained in literature is vast.

Examples of continuous-time queueing systems for which the decomposition property holds include Borst[1] (different polling models), Loris-Teghem[12] (waiting time in exhaustive vacation model), and Fuhrmann[6] (generalized vacations). This last paper proves the decomposition property for a large class of continuous-time queueing systems. In discrete-time queueing systems with vacations, a similar

2000 Mathematics Subject Classification. Primary: 60K25, 68M20; Secondary: 90B22.
Key words and phrases. Queueing theory, Decomposition, Vacations.
decomposition property holds; although mentioned in Takagi[15], it is not proven there. As we assume generalized vacations, our proof provides the discrete-time counterpart to the proof in Fuhrmann[6]. However the conditions under which the proof holds are more general, in the sense that the Poisson arrival process in Fuhrmann[6] is replaced by a batch Bernoulli arrival process.

In the next section we will describe a discrete-time queueing model with generalized vacations, and discuss the conditions needed for stochastic decomposition, which we prove in section 3. In section 4 we discuss four different types of discrete-time queueing systems for which the decomposition property holds. The first two are illustrations that can be found in literature, whereas the third and fourth present new examples, and results. Finally, in a last section, we examine some of the more limiting conditions needed for the given proof, and the possibility of relaxing or dropping these is explored.

2. Setup. Before focussing on the proof of the decomposition result, we list the conditions under which the proof holds. Assume a single-server queueing system with an infinite queue in discrete-time, in which customers arrive according to a batch Bernoulli arrival process (BBP). We write the number of customers entering the system during the $n$'th slot as $a_n$ and since the number of arrivals are i.i.d. from slot to slot, its probability generating function (pgf) $A(z) \triangleq E[z^{a_n}]$ is independent of $n$. Service times are i.i.d. (independent and identically distributed), and customers are served in an order that is independent of their service time. Balking, defecting, or reneging of customers is not allowed. Generalized vacations can interrupt service, however, such interruptions are nonpreemptive. That is, once selected for service, a customer is served to completion in a continuous uninterrupted manner. The decision whether or not a vacation starts or ends at a time $t$ must be made independently of the number of arrivals in the system during any slot after time $t$ - i.e. the vacation policy does not anticipate future jumps in the arrival process. Without loss of generality, we can assume that vacations last only one slot, and a vacation can be started immediately after the end of a previous vacation. The original decomposition property introduced in Fuhrmann[6] (for continuous time) required the additional assumption that the number of customers arriving during separate vacations are i.i.d. random variables. This assumption is met in the discrete-time counterpart studied here by assuming one-slot vacations and batch Bernoulli arrivals.

Furthermore, we assume that the system is stable in the sense that stationary distributions for the queue content at the beginning of a random vacation, at random departure instances and at the beginning of a random slot all exist. Finding stability conditions for a vacation system can be non-trivial, and a different stability analysis is required for each vacation system, which can vary in difficulty.

When stability is proven for this vacation system, stability will be ensured for the system without vacations, i.e. with the same arrival process and service time distribution, but without service vacations. We will call this system in the following, the simple queueing system, and it will play a major role in our decomposition result. Notice that the simple queueing system is a special vacation system in which the server can only go on a (one-slot lasting) vacation when the server is idle.
Decomposition Result. In this section, we provide a proof for the stochastic decomposition property in discrete time. Let us adopt the following definitions and conventions. First, an inactive slot is a slot during which the server is idle (no customers in the queue) or unavailable.

We say that customer $A$ is a root customer if customer $A$ entered the system during an inactive slot - i.e. not during the service time of another customer (see also Figure 1). Furthermore, customer $A$ is called an ancestor of customer $B$ when customer $B$ arrived in the system during the service time of customer $A$, or when customer $B$ entered the system during the service time of customer $C$ where customer $A$ is an ancestor of customer $C$. Note that in the latter case customer $C$ is an ancestor of customer $B$, making 'ancestry' a transitive relation. It is easy to see that each non-root customer has exactly one ancestor that is also a root customer; we call this customer the former customer’s root. We adopt the convention that a root customer is its own root, such that each customer entering the system has a root.

If customer $A$ (respectively $B$) is the root of customer $a$ (respectively $b$) and customer $A$ and customer $B$ entered the system during the same inactive slot, we say that customer $a$ and customer $b$ are part of the same ancestral line. This relation is an equivalence relation and we call each equivalence class an ancestral line.

Let $e$ be the stationary number of customers left in the vacation system by a random departing customer, that are also in the departing customer’s ancestral line. We will show that $e$’s distribution is independent of the vacation policy, (i.e. when the server becomes unavailable) as long as the number of customers entering the system during an inactive slot is unaffected by the vacation policy. Notice that if service is exhaustive, - i.e. a vacation starts after a departure only when the system is empty (see e.g. p.93 in Takagi[15]) - the system simplifies to the simple queue, since vacations last only one slot. For this simple queue, $e$ represents the total queue content (at departure epochs), since the ancestral line of a random customer, is then equal to the set of all customers served in the busy period during which this customer was served.
Choose a random (tagged) customer and let $A$ be its ancestral line. Let $A_k$ with $k \in \{1, \ldots, |A|\}$ be the subset of customers of $A$ left in the system by the departure of the $k$'th customer in $A$. When the tagged customer is the $l$'th customer of its ancestral line to be served, then $e = |A_l|$. Notice that the number of customers of $A$ in the system can only change during service times of customers that belong to $A$. Hence we find that

$$|A_{k+1}| = |A_k| - 1 + \sum_{n=1}^{s_{k+1}} a_{k+1,n}, \; k \in 1, \ldots, |A| - 1, \tag{1}$$

where $s_{k+1}$ is the service time of the $k+1$'st customer of $A$ to be served, and $a_{k+1,n}$ the number of customers entering the system during the $n$'th slot of the service time of the $k+1$'th customer in $A$. Furthermore we have that

$$|A_1| = r - 1 + \sum_{n=1}^{s_1} a_{1,n}, \tag{2}$$

where $r$ represents the number of root customers in $A$. Its pgf is well-known and given by $E[z^r] = \frac{1}{E[z]} \frac{dA(z)}{dz}$. Lastly it is trivial to see that $A_{|A|}$ is the first and only set in the series $(A_1, A_2, \ldots)$ to be empty. With $S(z)$, pgf of a random customer’s service time, and using the above observations, we find that $E(z)$, pgf of $e$ can be calculated yielding

$$E(z) = (1 - A'(1)S'(1)) A(z) - 1 \frac{A(z) - 1}{A'(1)(z - 1)} \frac{(z - 1)S(A(z))}{z - S(A(z))}. \tag{3}$$

This pgf $E(z)$ was calculated without any additional assumptions on the vacation policy (other than not anticipating future jumps in the arrival process). Also note that the pgf of the number of customers left by a random departing customer in the simple queueing system is also given by the above expression (see e.g. Bruneel[3]).

Let $d$ be the stationary number of customers left in the vacation system by a random (tagged) customer, and let $x$ be the stationary number of customers in the vacation system at the beginning of the arrival slot of this tagged customer’s root. Here, the service order does not affect the stationary distribution of the queue content at for instance the beginning of a random slot or after a random departure (among others), as long as this order is independent of customers’ service times. Thus without loss of generality, we may consider a LIFO service discipline. With this service discipline, upon departure of the tagged customer, the only customers in the queue will either have been waiting in the queue at the beginning of the arrival slot of the tagged customer’s root (none of them has left because of the LIFO service discipline) or in the tagged customer’s ancestral line. Of the former, there are exactly $x$, and of the latter, there are exactly $e$. Because a customer in the ancestral line of the tagged customer cannot be in the queue at the beginning of the
arrival slot of the tagged customer’s root, \( x \) and \( e \) have no customers in common, and we have that \( d = x + e \). Moreover, because of the BASTA-property (see for instance Halfin\cite{7}), \( x \) and \( e \) are independent random variables. This means the pgf of \( d \) is the product of those of \( x \) and \( e \).

Let \( f \) be the number of customers that arrived in the vacation system during the arrival slot of our tagged customer that are served before it. An unpublished theorem by Burke from 1968 can be found in Cooper\cite{4} (p.187), stating that when a stationary distribution exists for \( d \) (which we assumed), the stationary distribution for the number of customers found by a random arriving customer in the system also exists, and these distributions are the same (a similar result was mentioned in Kleinrock\cite{8}). The theorem presupposes single arrivals and single departures. The single departures are ensured by the limitation of one server. The single arrivals can be forced by assuming that customers that enter the system during the same slot (in batch) observes (at its arrival instant) only the fraction of this batch that is to be served before him. Hence, \( d \) is distributed as the sum of \( v \), stochastically equivalent to the number of customers in the vacation system at the beginning of a random customer’s arrival slot, and \( f \), i.e., \( d = v + f \). The random variable \( f \) itself has a pgf given by the following expression.

\[
F(z) \equiv E[z^f] = \frac{A(z) - 1}{A'(1)(z - 1)}.
\] (4)

Since our last observation \( (d = v + f) \) is valid for an arbitrary vacation system, it is also true for the simple queueing system. Namely that \( e \) is stochastically equivalent to the number of customers in this simple queueing system seen by a random arriving customer (assuming the single arrivals from previous paragraph). Furthermore, it is distributed as the sum of \( u \), the number of customers in the simple queueing system at the beginning of the arrival slot of a random customer, and \( f \) (see for instance Bruneel\cite{3}), i.e., \( e = u + f \). Whether our observation epoch for \( v \) and \( u \) is the beginning of the arrival slot of a random customer or the beginning of a random slot does not impact their distribution since we assume a BBP and so the BASTA-property holds. The same is true for \( x \) - i.e. whether we observe the vacation system at the beginning of the arrival slot of a random customers’ root, or at the beginning of a random inactive slot makes no difference. Summarizing we have:

\[
v + f = d = x + e = x + (u + f) \quad \Rightarrow \quad v = x + u.
\] (5)

In words: The number of customers in the system with generalized vacations at the beginning of a random slot is distributed as the sum of the number of customers in this system at the beginning of a random inactive slot and the number of customers in the simple queueing system at the beginning of a random slot.
A somewhat counter-intuitive consequence of this decomposition property is that regardless of the load and vacation policy, the mean number of customers in the system at the beginning of a random slot is greater than the mean number of customers in the system at the beginning of an inactive slot. For high workloads (little idle time) this basically comes down to saying that the system occupancy is (on average) smaller during blocked periods (during which no messages leave the system) than it is on average.

4. Applications. The above decomposition result can be applied to numerous different systems where the server is not always available for some class of customers. The most classical example is a system with multiple types of customers of which three examples will be given. The decomposition property greatly simplifies the analysis of the last of these, while the first two are for illustration purposes (non-preemptive priority, polling systems). Another less obvious example concerns optical buffering, using FDLs. New results are mentioned under this example.

4.1. Example 1: Non-preemptive Priority. We start by analysing a simple example for which results can be found in literature (e.g. Walraevens[17]), in order to illustrate the use of the decomposition property. This first example concerns a discrete-time queueing system in which customers pertaining to two different priority classes enter and are served in an order dictated by the absolute non-preemptive priority rule. Arrivals are governed by a batch Bernoulli process and the number of high and low priority customers entering the system during the same slot may be correlated. We denote their joint pgf by $A(z_1, z_2) = E[z_1^{a_1} z_2^{a_2}]$, in which $a_j$ is the number of type-$j$ customers entering the system during a random slot (type-1 customers have priority over type-2 packets). Service times are i.i.d. and may be generally distributed. Service times of high and low priority customers may be distributed differently however, and we use $S_j(z)$ to represent the pgf of the service time of a random type-$j$ customer.

When one is concerned with obtaining the pgf of the number of type-1 customers in the system at the beginning of a random slot (later called $V_1(z)$), the decomposition property applies. Concretely, we view slots during which a type-2 customer is being served as vacation slots (no type-1 customer is being served). We obtain that $V_1(z)$ is the product of on the one hand, the pgf of the number of type-1 customers in the system at the beginning of a random inactive slot (idle slot or slot during service of type-2 customer), which we will denote by $X_1(z)$, and on the other hand, the pgf of the number of type-1 customers in the associated simple queueing system, namely without type-2 customers entering the system, at the beginning of a random slot (which we’ll denote by $U_1(z)$).

The latter pgf can be found in literature (see for instance Bruneel[3]):

$$U_1(z) = (1 - \rho_1) \frac{(z - 1)S_1(A(z, 1))}{z - S_1(A(z, 1))}, \tag{6}$$
STOCHASTIC DECOMPOSITION AND ITS APPLICATIONS

Figure 2. The timeline zooms in on the service time of customer $C$ (see paragraph below (7)). The type-1 customers in the queue at the start of the randomly chosen inactive slot, are those that entered during the $r$ slots preceding it.

where $\rho_j$ is the fraction of time a type-$j$ customer occupies the server - i.e. $\rho_1 = \frac{d}{dz} S_1(A(z, 1)) |_{z=1}$. The pgf $X_1(z)$ envelopes the nature of non-preemptive priority, since $U_1(z)$ contains no information about the specifics of this priority system. Because the randomly chosen inactive slot is an idle slot with probability $1 - \rho_1$ - where $1 - \rho = 1 - \rho_1 - \rho_2$ is the probability a random slot is an idle slot - we condition $X_1(z)$ as follows:

$$X_1(z) = \frac{1 - \rho}{1 - \rho_1} + \frac{\rho_2}{1 - \rho_1} X_1^*(z), \quad (7)$$

in which $X_1^*(z)$ represents the pgf of the number of type-1 customers in the system at the beginning of a random slot during which a type-2 customer is being served. Let that type-2 customer be called customer $C$. At the beginning of customer $C$’s service time, no type-1 customers were in the queue, otherwise customer $C$ would not have been elected for service (see Figure 2). By a renewal-type argument we therefore obtain:

$$X_1^*(z) = \frac{S_2(A(z, 1)) - 1}{S_2'(1)(A(z, 1) - 1)}. \quad (8)$$

We can obtain $V_1(z)$ as the product of $X_1(z)$ and $U_1(z)$. The same decomposition result also applies for the distribution of the number of type-2 customers in the system at the beginning of a random slot (pgf $V_2(z)$). Although this time finding $X_2(z)$ proves to be more involved - $X_2(z)$ is the pgf of the number of type-2 customers at the beginning of a random inactive slot (idle slot or slot during which a type-1 customer is being served). However from the results found for $V_2(z)$ in for example Walraevens[17] we can deduce that

$$X_2(z) = \frac{V_2(z)}{U_2(z)} = \frac{1 - \rho}{1 - \rho_2} \frac{z - S_2(A(1, z))}{A(1, z) - 1} \frac{A(Y_1(z), z) - 1}{z - Y_2(z)}. \quad (9)$$
The pgf $Y_1(z)$ is implicitly defined by $Y_1(z) = S_1(A(Y_1(z), z))$ and $Y_2(z) \triangleq S_2(A(Y_1(z), z))$.

Although in the case of $V_1(z)$, the decomposition property proved to be a useful tool to ease the analysis, this is not the case for type-2 customers. A direct calculation of $V_2(z)$, which can be found in [17], proves to be of the same complexity as the calculation of $X_2(z)$. In this case, the decomposition property can provide $X_2(z)$ more readily than direct analysis. Likewise, if one were interested in the pgf of the number of type-$j$ customers in the system at the beginning of a slot during which a type-$j$ customer is being served, an answer can be easily formulated as a function of $X_j(z)$.

4.2. Example 2: Polling Systems. The decomposition result can be applied to many different polling systems (see for instance [1] and [12] in continuous time). Here we show how it is applied in obtaining the pgf of the number of customers at station $i$ in a polling system with $N$ stations and exhaustive-service, at the beginning of a random slot. It is only necessary for the arrivals at station $i$ to follow a batch Bernoulli process, and the service times of customers at this station to be iid. For this example we assume service times of one slot, and i.i.d. distributed switchover times. Let $A_i(z)$ be the pgf of the number of customers joining station $i$ during a random slot (with $\lambda_i$ as its mean), and $F_i(z)$, pgf of the number of customers at station $i$ at a random polling instant of station $i$. Takagi[16] (on page 63) tells us that $V_i(z)$, pgf of the number of customers at station $i$ at the beginning of a random slot is given by following expression:

$$V_i(z) = (1 - \lambda_i) \left( 1 + \frac{A_i(z) - 1}{z - A_i(z)} \right) \frac{p}{1 - \lambda_i} \frac{1 - F_i(z)}{1 - A_i(z)},$$

in which $p$ is the probability that a random slot is a polling instant at station $i$ - i.e. the inverse of the cycle time. Clearly, the first two factors in this pgf represent the pgf of the number of customers in the simple queueing system (one station, no switchover times). Therefore the last two terms form a pgf, which is precisely the pgf of the number of customers at station $i$ at the beginning of a random slot during which the server is not serving a customer at station $i$.

4.3. Example 3: Slot-Bound Priority. In [5] a discrete-time queueing system was analysed with a batch Bernoulli arrival process of customers of two classes. These customers have general class-dependent service times (as in the previous example), and are served in the order in which they joined the buffer. Customers that entered the buffer during the same slot are served in order of their priority class (slot-bound priority) - i.e. the type-1 customers are served before the type-2 customers. Let $A(z_1, z_2)$, $S_j(z)$, $\rho_j$, and $\rho$ be defined as in example 1, then the decomposition property can be applied to obtain the pgf of the number of type-1 customers in the queue at the beginning of a random slot, greatly simplifying the analysis in [5] in the case were we are interested in the marginal stationary queue content distribution.
Figure 3. The type-1 customers in the system at the beginning of slot $I$, are accumulated in the queue during the $t_1$ time slots it takes customer $c$ to leave the system.

Again the strategy is to view slots during which a type-2 customer is being served as vacation slots. Since we already know $U_1(z)$, the pgf of the number of type-1 customers in the associated simple queueing system at the beginning of a random slot (given in (6)), the remaining pgf to be determined is that of the number of type-1 customers in the system with slot-bound priority at the beginning of a random inactive slot. We can again make the distinction between idle slots and vacation slots. The unknown pgf $X_1^*(z)$ then represents the pgf of $x_1^*$, the number of type-1 customers in the system at the beginning of a random vacation slot.

Let the type-2 customer we are serving during this random (tagged) vacation slot be called customer $C$, and the set of customers that entered the system during the same slot as customer $C$, the base. Note that at the beginning of the tagged slot all type-1 customers in the base will have left the system because of the slot-bound priority rule. Hence, all type-1 customers in the system at the beginning of this tagged slot must have arrived after customer $C$. Let $t_1$ be the number of slots between the arrival slot of customer $C$ and the tagged slot (see Figure 3). The relation between $t_1$ and $x_1^*$ then reads as follows:

$$x_1^* = \sum_{n=1}^{t_1} a_{1,n},$$

(11)

in which $a_{1,n}$ represents the number of type-1 customers arriving during the $n$th slot counted from the end of customer $C$’s arrival slot. Let $w$ be the work in the system at the beginning of customer $C$’s arrival slot - its distribution is the same as the work in the system at the beginning of a random slot, due to the BASTA-property. Then $t_1$ can be divided into two disjunct and independent time periods. First, $(w-1)^+$ where we use the notation $x^+ \triangleq \max(0,x)$, represents the delay customer $C$ suffers caused by customers in the queue prior to its arrival in the system. Its pgf can be found in literature (e.g. in Bruneel[3]):

$$W^-(z) \triangleq \E[z^{(w-1)^+}] = (1-\rho) \left(1 + \frac{A(S_1(z),S_2(z)) - 1}{z - A(S_1(z),S_2(z))}\right).$$

(12)

The second time period concerns a fraction of the combined service time of all customers in the base, namely the fraction preceding the tagged slot. This fraction
includes the service time of each type-1 customer in the base, because of the slot- 
bound priority paradigm. Following a renewal-type argument and keeping in mind 
the correlation between the number of type-1 and type-2 customers in the base, one 
can derive the pgf of \( r_1 \), the number of slots between the service initiation of the 
base and the tagged slot, with the following result:

\[
R_1(z) \triangleq \mathbb{E}[z^{r_1}] = \frac{A(S_1(z), S_2(z)) - A(S_1(z), 1)}{\rho_2(z - 1)}.
\]  

(13)

Using the fact that \( T_1(z) = W^{-1}(z)R_1(z) \) and \( X_1^t(z) = T_1(A(z, 1)) \) (which can 
be obtained from (11)), we have enough information to calculate \( X_1(z) \) and thus
\( V_1(z) \), leading to

\[
X_1(z) = \frac{1 - \rho}{1 - \rho_1} \frac{A(z, 1) - A(S_1(A(z, 1)), 1)}{A(z, 1) - A(S_1(A(z, 1)), S_2(A(z, 1)))}
\]

\[
V_1(z) = (1 - \rho) \left( 1 + \frac{S_1(A(z, 1)) - 1}{z - S_1(A(z, 1))} \right) \frac{A(z, 1) - A(S_1(A(z, 1)), 1)}{A(z, 1) - A(S_1(A(z, 1)), S_2(A(z, 1)))}.
\]  

(14)

Obtaining an expression for \( V_2(z) \) is, contrary to example 1, an easier task and 
quite analogous to the above procedure, the only difference being that the base 
will contain all type-2 customers originally in it. As these examples show, the 
decomposition property can be used to effectively simplify the analysis of this kind 
of system, that would otherwise require a more extensive analysis.

4.4. Example 4: FDL buffer. All previous examples concerned multiclass 
queueing systems, in which the service times of non-interesting customer types 
were labeled as vacations. The second example for instance uses a combination of 
other customer types’ service times and genuine vacations in the form of switchover 
times as inactive slots. Introducing a new result, this examples focusses on a single-
class queue with vacations that depend on a combination of the arrival instant of 
the customer to be served after the vacation, and the queue content at that arrival 
instant.

Optical buffers contain a set of fiber delay lines (FDLs), assigning a fixed delay to 
the packets (customers) sent through it. To prevent collisions, packets are routed 
to different FDLs, often creating voids during which no packets are sent, simply 
because the exact delay needed in order not to collide is typically not available, and 
a (larger) delay line has to be chosen. As of yet, there are no general results for the 
queue content of such a system, though there are results for very specific optical 
buffer systems (e.g. Lakatos[10], for the case of a continuous-time \( M/M/1 \) FDL 
buffer). A lot of work has been done on the analysis of delay (waiting time) and loss 
(see for instance Rogiest[13] with derivation of the entire delay distribution), while 
the system content remains largely unstudied. When the arrival process is Bernoulli 
and the service times are generally distributed and i.i.d., the decomposition property
can be applied to the system content distribution.

More specifically we let $B(z)$ denote the pgf of the burst-size and consequently the service time of a random burst, and let $A(z)$ be the number of bursts (one or none) entering the system during a random slot – arrivals occur i.i.d. from slot to slot. The optical buffer consists of an infinite number of delay lines, each of which give a fixed delay to an incoming burst routed through it. The available delays given by these FDLs are all integer multiples of the granularity being $D$ time slots. Only one burst can enter the system each slot (Bernoulli arrival process) and upon arrival it introduces a so-called void in the system. These voids are times during which the server is not processing bursts even though the queue is not empty, and are inherent to the switching mechanism. The chosen FDL for the burst to be routed to, is the FDL that causes the least amount of delay, such that the burst is served after all bursts present in the system at the time of its arrival – FCFS, so no void filling – and its service does not start during the service time of a previous burst (no collisions). This model has been studied first in Laevens\cite{Laevens}, in which the pgf of the waiting time and sojourn time of a random burst were derived – $W(z)$ and $G(z)$ respectively. There is an easy relation between the two namely:

$$G(z) = W(z) B(z).$$

4.4.1. Optical Buffer Content. The optical buffer system can be seen as a queueing system with vacations, in which the voids play the role of vacations. $X(z)$ then represents the pgf of the number of bursts in the system at the beginning of a random inactive slot – meaning an idle slot or a slot during a void period (later called 'void slot'). The question to be answered is whether we can find $V(z)$, pgf of the number of bursts in the system at the beginning a random slot, through $X(z)$, using the decomposition property.

In order to obtain $X(z)$ we need to introduce an auxiliary discrete random variable (drv). When a burst enters the system, the system may already contain some other bursts that will be served before it. The time needed for those bursts to leave the system is called the scheduling horizon – $H$ with pgf $H(z)$; see also Laevens\cite{Laevens}. To respect FCFS scheduling, the burst has to wait for at least this amount of time. Naturally since the available FDLs provide delays that are multiples of $D$, $W$, the delay of this random burst is the first integer multiple of $D$ larger or equal to $H$, or

$$W = \lceil \frac{H}{D} \rceil D.$$

The void created by such an arrival consequently equals $W - H$ time slots. To calculate $X(z)$ we naturally make the distinction between whether the random inactive slot is an idle slot or a void slot. The amount of bursts in the system at the beginning of an idle slot is 0 while the probability that an inactive slot is an
idle slot is $\frac{V(0)}{1-\rho}$. The workload $\rho$ here is $A'(1)B'(1)$. Due to the occurrence of voids, the maximum tolerable load under which the system is stable is smaller than 1. A detailed study is provided in Rogiest[14].

The set of bursts in the system at the beginning of a random void slot (later called slot $I$) contains the burst $B_0$ that created the void period in which slot $I$ lies, as well as all bursts that arrived after $B_0$ but before the beginning of slot $I$ – see also Figure 4. The period during which these bursts arrived equals the scheduling horizon of $B_0$ plus a fraction of the void that $B_0$ created upon arrival, namely the fraction of this void up until but not including slot $I$. Let $H^*$ be $B_0$’s scheduling horizon. Its distribution is not the same as the scheduling horizon of a random burst, – a random void slot was selected and not a random burst – hence the notation $H^*$ instead of $H$. Likewise we introduce $Q$ as the part of the void period introduced by $B_0$ before slot $I$. With $X_v$ the number of bursts in the system at the beginning of slot $I$, and $A_n$ the number of bursts arriving in slot $n$ counted from the slot following $B_0$’s arrival, we find that

$$X_v = \sum_{n=1}^{H^*+Q} A_n + 1. \quad (17)$$

The ‘+1’ at the end represents burst $B_0$ – remember that the presence of a void implies the presence of at least one burst. To calculate the joint pgf of $H^*$ and $Q$ and later the pgf of $H^* + Q$, we notice that

$$\begin{align*}
\Pr[H^* = k, Q = m] &= \Pr[H^* = k] \Pr[Q = m | H^* = k] \\
\Pr[H^* = k] &= \frac{[k]_D - k}{E[[H]_D - H]} \Pr[H = k] \\
\Pr[Q = m | H^* = k] &= \frac{1}{[k]_D - k}, \text{ if } 0 \leq m < [k]_D - k,
\end{align*} \quad (18)$$

![Figure 4](image.png)
where \(\lceil k \rceil_D\) denotes the smallest integer multiple of \(D\) larger than \(k\), \(\lfloor k/D \rfloor\). Also note that \(\lceil k/D \rceil = W\). Consequently we find that

\[
E[y^Q z^{H^*}] = \frac{E[z^H \sum_{m=0}^{W-H-1} y^m]}{E[W - H]} = \frac{E[z^H (y^{W-H} - 1)]}{E[W - H](y - 1)},
\]

(19)

\[
E[z^{H^*+Q}] = \frac{W(z) - H(z)}{E[W - H](z - 1)}.
\]

(20)

Using (17) and (20), the pgf of \(X_v\) is found, as

\[
E[z^{X_v}] = \frac{W(A(z)) - H(A(z))}{E[W - H](A(z) - 1)}.
\]

(21)

The probability that a random inactive slot is a void slot is \(1 - \rho - V(0)\), and so \(X(z)\) can be found using the following relation.

\[
X(z) = \frac{V(0)}{1 - \rho} + \frac{1 - \rho - V(0)}{1 - \rho} E[z^{X_v}].
\]

(22)

We already know that the simple queue (no voids) has a system content with pgf \(U(z)\) which is given by (see e.g. Bruneel[3])

\[
U(z) = (1 - \rho)(z - 1)B(A(z)) \quad z - B(A(z)).
\]

(23)

The decomposition property then yields \(V(z) = U(z)X(z)\). When we use the fact that \(E[W - H](1 - A(0)) = 1 - \rho - V(0)\) – the left hand side is the average void introduced by a potentially arriving burst (0 if non arrive) – and a relation between \(H(z)\) and \(W(z)\) that can be found in Laevens[9] (see below), we can simplify \(V(z)\) to find a very elegant result.

\[
H(z) = \frac{1 - A(0)}{z - A(0)} W(z)B(z) + V(0) \frac{z - 1}{z - A(0)}.
\]

(24)

\[
V(z) = G(A(z)).
\]

(25)
Indeed this is no coincidence. In contrast to previous examples, a direct approach works better in this case. After all, the decomposition property gives us an alternative method to calculate $V(z)$, not necessarily a shorter method. The direct approach goes as follows. The distribution of $v$ is the same as that of the number of bursts in the system just prior to the arrival of a random burst in the system since bursts arrive according to a Bernoulli arrival process (BASTA-property). Moreover because of the same theorem by Burke used in section 3 it also has the same distribution as the number of bursts left in the system by a random departing burst. The bursts queued in the optical buffer at that time are those that arrived after the departing burst (FCFS service discipline), and before its departure. This time period is precisely the delay of this departing customer and since the customer was randomly chosen its delay is given by $W + B$. In effect the queue content is then given by:

$$v = \sum_{n=1}^{W+B} A_n.$$  \hfill (26)

From this relation and (15) we see that (25) follows rather directly.

4.4.2. Unfinished Work in Optical Buffer. It is not difficult to see that the decomposition property also applies to the unfinished work at the beginning of a random slot, taking into account that a 'customer' now represents one unit of work and the service time of such a customer is one time slot. At a certain time, the unfinished work represents the number of slots required for the system to serve all bursts present in the system, without counting the void slots (during which no bursts are served). A direct analysis, similar to the one followed in the previous paragraph to obtain $V(z)$, can also be performed to obtain the pgf for the unfinished work at the beginning of a random slot. A complication here is that the BASTA property cannot be used directly.

The purpose of this paragraph is to show an alternative use for the decomposition property. Combining the given that customers are served in a continuous way without interruptions, and the fact that all service times are i.i.d., we see that at the beginning of an inactive slot the unfinished work in the system consists of the sum of a set of i.i.d. drv’s, namely the service times of the bursts in the system. The number of bursts in the system at the beginning of a random inactive slot is characterized by pgf $X(z)$, and hence the unfinished work in the system at that time has a pgf given by $X(B(z))$.

To express the decomposition property for the unfinished work, we introduce following notation. Let $\hat{V}(z)$ denote the pgf of the unfinished work in the system at the beginning of a random slot, $\hat{U}(z)$ the pgf of the unfinished work in the simple queue, and $\hat{X}(z) = X(B(z))$ the pgf of the unfinished work in the system at the beginning of an inactive slot. Then, the decomposition states that $\hat{V}(z) = \hat{U}(z)\hat{X}(z)$ (see also Boxma[2], and p.92 in Takagi[15]). An expression for $\hat{U}(z)$ can be found in literature (see e.g. Bruneel[3]):
\[ \hat{U}(z) = (1 - \rho) \frac{(z - 1)C(z)}{z - C(z)}, \]  
(27)

where \( C(z) = A(B(z)) \) is the pgf of the amount of work entering the system each slot – which is i.i.d. from slot to slot. By virtue of combining (25) with the decomposition property applied to bursts, the expression for \( X(z) \) can also be written as follows.

\[ X(z) = \frac{V(z)}{U(z)} = \frac{(z - B(A(z)))W(A(z))}{(1 - \rho)(z - 1)}. \]  
(28)

The expression for \( \hat{V}(z) \) follows directly from \( \hat{X}(z) = X(B(z)) \) and the decomposition property applied to unfinished work:

\[ \hat{V}(z) = W(C(z)) \frac{(z - 1)C(z)}{z - C(z)} \frac{B(z) - B(C(z))}{B(z) - 1}. \]  
(29)

5. **Extensions.** The decomposition property, although applicable for a wide variety of discrete-time queueing systems, does have limitations, which we will now briefly discuss.

**Preemptive vacation policies** would include those vacation policies that allow a customer’s service time to be interrupted by a vacation. Since customers are not served in a continuous uninterrupted manner, the decomposition result can not be applied directly. However it is not easy to see why the employed arguments would fail in case of preemptive vacation policies. In case of non-preemptive vacations, the service order does not affect \( d \)'s distribution. The choice of what customer to serve at service initiation instants, if it is made independently of the customer's service times, does not affect \( d \)'s distribution because of the assumption of i.i.d. service times. When service is preempted, \( x \), the number of customers in the system at the beginning of the arrival slot of a random (tagged) customer’s root, includes the customer that was in service when the vacation started. Whether or not this preempted customer leaves the system before the tagged customer, - i.e. by resuming its service time directly after the vacation - directly affects \( d \)'s distribution. This is the main reason why preemptive vacations are excluded from the analysis. In future research we hope to formulate exceptions to this rule such that the decomposition result (possibly in some modified form) can still be applied for preemptive vacations.

**Balking, reneging, and defecting** of customers is not permitted, because unlike server vacations, these impatient events directly affect the queue content. When however a general decomposition theory should exist which allowed balking/reneging/defecting, then it would need to constrain these events in some similar fashion as the current decomposition theorem constrains vacations - i.e. they can...
Correlated arrivals pose a genuine problem to the decomposition property, as $e$ will become dependent on the vacation policy itself. Hence no general decomposition theorem can be derived in the same way as Fuhrmann[6]. Even when tracking the phase in case of DB-MAPs, the dependency on the vacation policy can only be thwarted in some very specific cases.

6. Conclusion. This paper assumes a discrete-time $Geo^X/G/1$ queueing system with generalized vacations. The conditions needed for stochastic decomposition of the queue content at the beginning of a random slot were explored, and the decomposition property was established and discussed in detail. Possible extensions questioning these very conditions (such as preemptive vacations) were briefly discussed. Furthermore, some examples, ranging from multi-class queues, to polling systems, and even including optical buffers, show the wide variety of applications this decomposition result enjoys. As a consequence new results were obtained concerning the buffer content of an FDL buffer system.

REFERENCES

[1] S.C. Borst, O.J. Boxma Polling Models with and without Switchover Times, Operations Research, 45 (1997), 536-543.
[2] O.J. Boxma, W.P. Groenendijk Pseudo-conservation laws in cyclic-service systems, Journal of Applied Probability, 24 (1987), 949-964.
[3] H. Bruneel Performance of Discrete-Time Queueing Systems, Computers Operations Research, 20 (1993), 303-320.
[4] R.B. Cooper ’Introduction to Queueing Theory, Ed.2’, North-Holland (Elsevier), New York (1981).
[5] S. De Clercq, B. Steyaert and H. Bruneel Analysis of a Multi-Class Discrete-time Queueing System under the Slot-Bound Priority rule, Proceedings of the 6th St. Petersburg Workshop on Simulation, (2009), 827–832.
[6] S.W. Fuhrmann, R.B. Cooper Stochastic Decompositions in the M/G/1 Queue with Generalized Vacations, Operations Research, 33 (1985), 1117–1129.
[7] S. Halfin Batch Delays Versus Customer Delays, The Bell System Technical Journal, 62 (1983), 2011–2015.
[8] L. Kleinrock “Queueing Systems, Volume I: Theory”, Wiley, New York (1975)
[9] K. Laevens, H. Bruneel Analysis of a Single-Wavelength Optical Buffer, Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, (2003). 1–6.
[10] L. Lakatos Cyclic-Waiting Systems, Cybernetics and Systems Analysis, 46 (2010), 477-484.
[11] Z. Liang, S. Xiao Performance Evaluation of Single-Wavelength Fiber Delay Line Buffer With Finite Waiting Places, Journal of Lightwave Technology, 26 (2008), 520–527.
[12] J. Loris-Teghem On a Decomposition Result for a Class of Vacation Queueing Systems, Journal of Applied Probability, 27 (1990), 227–231.
[13] W. Rogiest, J. Lambert, D. Fiems, B. Van Houdt, H. Bruneel, C. Blondia A unified model for synchronous and asynchronous FDL buffers allowing closed-form solution, Performance Evaluation, 66 (2009), 343–355.
[14] W. Rogiest, K. Laeves, J. Walraevens and H. Bruneel Analyzing a Degenerate Buffer with General Inter-Arrival and Service Times in Discrete-Time, Queueing Systems, 56 (2007), 203–212.
[15] H. Takagi “Queueing Analysis, Volume 3: Discrete-Time Systems”, North-Holland (Elsevier), Amsterdam (1991)
[16] H. Takagi “Analysis of Polling Systems”, The MIT Press, Cambridge (Massachusetts 1986)
[17] J. Walraevens, B. Steyaert, H. Bruneel Performance analysis of the system contents in a discrete-time non-preemptive priority queue with general service times, Belgian Journal of Operations Research (JORBEL), 40 (2001), 91–103.

Received xxxx 20xx; revised xxxx 20xx.

E-mail address: Sofian.DeClercq@telin.ugent.be
E-mail address: Wouter.Rogiest@telin.ugent.be
E-mail address: Bart.Steyaert@telin.ugent.be
E-mail address: Herwig.Brunnel@telin.ugent.be