

# A queueing model of a foreground-background time-sharing system with general service times and constant swap times

**Citation for published version (APA):**

Schoot, van de, J. C. A. (1976). *A queueing model of a foreground-background time-sharing system with general service times and constant swap times*. (Memorandum COSOR; Vol. 7613). Technische Hogeschool Eindhoven.

**Document status and date:**

Published: 01/01/1976

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics

PROBABILITY THEORY, STATISTICS AND OPERATION RESEARCH GROUP

MEMORANDUM COSOR 76-13

A queueing model of a foreground-background  
time-sharing system with general service times  
and constant swap times.

by

J.C.A. van de Schoot

Eindhoven, September 1976

The Netherlands

A queueing model of a foreground-background time-sharing system with general service times and constant swap times.

by

J.C.A. van de Schoot

Summary.

A time-sharing system is modelled by a foreground-background queue with Poisson arrivals and independent and identically distributed service times with a general distribution function  $H$ . Waiting times in system are analysed for jobs, which arrive in a stationary process, and the Laplace-Stieltjes transforms of their distribution functions are derived.

This is done for a non-preemptive and a preemptive-resume priority rule with several policies to deal with constant swap times, which have to be attached each time the processor attends another job.

Some remarks about extensions to models with  $K > 2$  queues are made.

## I. Introduction.

A simple implementation of a time-sharing system is the foreground-background system (FB-system), in which many users are competing for service in the single central processing unit (the CPU).

Jobs arrive according to a Poisson-process with rate  $\lambda$  and join a first-in-first-out (FIFO) "foreground" queue (FQ), where each job is processed maximally up to  $Q$  units of time (the "time-slice"). If a job finishes within  $Q$ , it leaves the system. Otherwise, it is interrupted and it joins the FIFO "background" queue (BQ), where it waits for further service (see figure I.1)

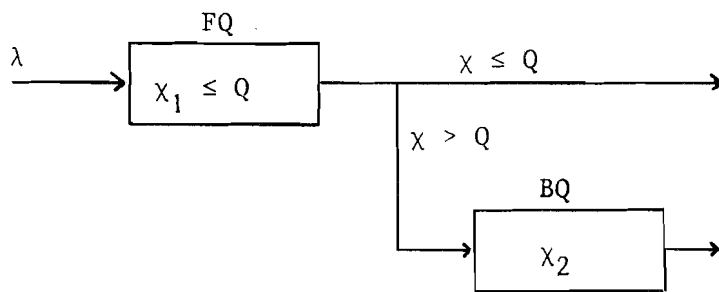


fig. I.1: an FB-system.

Fundamentally the FQ is higher in priority than the BQ. Two types of priority rules will be analysed.

### 1. The preemptive-resume priority rule.

Upon arrival jobs in the FQ interrupt jobs in the BQ that may be using the CPU. Hence, jobs in the BQ may run only when the FQ is empty. A job in the BQ resumes its processing from the point where it has been stopped.

### 2. The non-preemptive rule.

As in the previous case, jobs in the BQ start processing only when the FQ is empty. However, once a job from the BQ has been started, it is run to its end.

By this, the users who require a short processing time can be serviced relatively fast and independent of system loading; faster by use of the former than the latter rule.

Each job requires a service time  $\chi$  with general distribution function  $H$  independent of the others and of the arrival process. Since  $H$  is the distribution function of the computation time of jobs, we shall assume  $H$  to be continuous.

This model is also used by Guimaraes [6]<sup>1)</sup>. Similar models have been investigated with geometric (Kleinrock [7]) or exponential service times (Adiri and Avi-Itzhak [1], Coffman [3]). It is well-known, however, that service times in computer systems have large coefficients of variation (standard deviation over mean) and are better characterized by distributions like the hyperexponential (Fife [5]). In reference [11] pictures are given containing mean waiting times, deviations and coefficients of variation as functions of the time-slice  $Q$  for various rates  $\lambda$  and hyperexponential distribution functions  $H$ . These functions are derived from the work of Guimaraes. Although no general conclusions could be drawn, it becomes clear, yet, that it is not difficult to analyse an explicit FB time-sharing system with this model.

In this paper we will extend the above model by introduction of constant swap times. Each time the CPU attends another job, a constant swap time ( $S$ ,  $T$  or  $R$ ) will first be attached to its service time. When the job comes from the  $FQ$ , this swap time will be  $S$ ;  $T$  when it comes from the  $BQ$  for the first time and  $R$  when it comes from the  $BQ$  after preemption using the preemptive-resume priority rule. In chapter II the Laplace-Stieltjes transforms (LST's) belonging to the different waiting-time distributions in the system will be derived in the case of  $R = 0$  and for both priority rules. The case of  $R \neq 0$  using the preemptive-resume priority rule will be analysed in chapter III. For various policies, which attach  $R$  to the service time of a job, we will derive the LST belonging to the total waiting time in system by making use of the unfinished work functions and of methods as can be found in Kleinrock's work [8].

In both chapters some remarks about extensions to models with  $k > 2$  queues are made.

1) Repeatedly Guimaraes [6] is referred to, since this paper has mainly been inspired by his work.

## II. The model with constant swap times S and T.

In Guimaraes' model the CPU is processing jobs, exclusively. In a more realistic model of a time-sharing system we will have to take into account the other activities of the CPU, like: administration for identification and protection of the different jobs, for storing and getting the data of a job; data transport between primary and secondary memory making the data available for use by the CPU. Truly, these activities will have to be done on behalf of the treatment of the jobs, but the time spent on them depends strongly on the Operating System of the particular organization, in which the time-sharing system has been implemented. Since these activities must be carried out just at the change-over to another queue or to another job, we will enlarge the model by the introduction of the constant swap times S and T first.

The swap times will be assumed to be constant avoiding an unnecessarily more complicated model. However, by assuming independence between the swap time and all other stochastic variables in the model it would be easy to analyse the same model, in which the swap time would be stochastic.

Hence we assume that a constant swap time S is needed, each time the CPU starts processing a new job from the FQ. Furthermore, when a job from the BQ enters the CPU for the first - and maybe only - time, a constant swap time T is needed. An eventual resumption of a job from the BQ after an interruption using the preemptive-resume priority rule occurs immediately at the point where stopped (note that  $R = 0$  in this chapter).

We denote by  $p$  the probability that a job requires a service time of more than  $Q$  units of time and we define

$$N(\chi) = \begin{cases} 0 & \text{if } \chi \leq Q ; \Pr(\chi \leq Q) = 1 - p \\ 1 & \text{if } \chi > Q ; \Pr(\chi > Q) = p . \end{cases}$$

Then the total processing time  $\varphi$  - swap times S and T included - of a job with service time  $\chi$  equals  $\chi + S + N(\chi) \cdot T$  and its distribution function is  $\Phi$ .

Let  $\varphi_1$  be the total processing time - swap time S included - given to a job in the FQ with distribution function  $\Phi_1$ . Then

$$\Phi_1(x) = \begin{cases} \Phi(x) & \text{if } x \leq Q + S \\ 1 & \text{if } x > Q + S \end{cases}$$

Let  $\varphi_2$  be the total remaining processing time of a job - swap time T included - and its distribution function, given that  $\chi > Q$ , is  $\Phi_2$ .

Then  $\phi_2(x) = \Pr(\phi_2 \leq x \mid \phi > Q + S) = \frac{1}{p}[\phi(x + Q + S) - \phi(Q + S)]$ ,  
where  $p = 1 - H(Q) = 1 - \phi(Q)$ .

By  $\phi^*$ ,  $\phi_1^*$  and  $\phi_2^*$  we denote the LST's of  $\phi$ ,  $\phi_1$  and  $\phi_2$  respectively, where the LST of a distribution function  $F$  of  $x$  is defined as

$$F^*(s) = \epsilon e^{-sx} = \int_{-\infty}^{\infty} \text{EXP}[-sy] dF(y) \quad 2,3), \text{ Re}(s) \geq 0.$$

By  $\tau_r$ ,  $t_r$  and  $u_r$  we denote the  $r$ -th moment of  $\phi$ ,  $\phi_1$  and  $\phi_2$  respectively, where the  $r$ -th moment of  $x$  is defined as  $\epsilon x^r$ , for  $r \in \mathbb{N}$ .

By  $\rho_1$ ,  $\rho_2$  we denote the "utilization factor" of the system formed by the CPU and the FQ respectively the BQ, where the utilization factor of a system is defined as the ratio of the rate at which "work" enters the system to the maximum rate (capacity) at which the system can perform this work; the work an arriving job brings into the system equals the number of units of service time it requires. So, in the case of a single-server system, the definition of  $\rho$  becomes

$$\rho = (\text{average arrival rate of jobs}) \times (\text{average service time}) .$$

Hence  $\rho_1 = \lambda t_1$  and  $\rho_2 = \lambda p u_1$ .

Analyzing the waiting times we will study in all cases the processes under steady state conditions only.

Define

- $\eta_1$  = the waiting time in the FQ for a job arriving in a stationary process.
- $\eta_2$  = the total waiting time in the system plus the processing time in the FQ for a job arriving in a stationary process, which requires more than  $Q$  units of service time.
- $\eta_3$  = the waiting time in the BQ caused by interrupts of the own background service only for a job arriving in a stationary process, which requires more than  $Q$  units of service time.
- $\gamma_t$  = total time in the system (also called response time) conditioned on a given service time requirement  $t$  for a job arriving in a stationary process.

---

2):  $\epsilon$  is the expectation operator.

3): Especially when the exponent is complicated, we will use  $\text{EXP}[-x]$  in stead of  $e^{-x}$ .

$T_t$  = the distribution function of  $\gamma_t$ .  
 $T_t^*$  = the LST of  $T_t$ .  
 $T_{r/t}$  = the r-th moment of  $\gamma_t$  for  $r \in \mathbf{N}$

Further for  $i = 1, 2$  and  $3$

$W_i$  = the distribution function of  $\eta_i$ .  
 $\Omega_i$  = the LST of  $W_i$ .  
 $w_r^{(i)}$  = the r-th moment of  $\eta_i$  ( $r \in \mathbf{N}$ ).

Now, the above model looks like the model of Guimaraes replacing  $H, H_1, H_2$  and  $Q$  by  $\phi, \phi_1, \phi_2$  and  $(Q + S)$  respectively. Hence Guimaraes' results are available immediately.

Using the method of imbedded Markov chains, Guimaraes has found the stationary distributions and/or the LST of the queue size distribution (par. 3) and of the waiting time distributions (par. 4) in his model without swap times. And with them he computes the expectations of the various stochastic variables.

Note II.1.: The waiting time  $\eta_3$  and its contribution to  $\eta_2$  and  $\gamma_t$  doesnot occur in Guimaraes' work. Hence his derivations of  $\Omega_2$  and  $T_t^*(s)$  are incomplete in the case of the preemptive-resume priority rule (see par. 4).

Complete derivations are

$$\Omega_3(s) = \psi_2^*(\lambda - \lambda\Gamma(s));$$

$$w_1^{(3)} = \rho_1 b / (1 - \rho_1);$$

$$\Omega_2(s) = \Omega_3(s) \cdot \Omega(s + \lambda - \lambda\Gamma(s)) \cdot \text{EXP}[-Q(s + \lambda - \lambda\Gamma(s))];$$

$$W_2(x) = \sum_{j=0}^{\infty} \int_{y=0}^{\infty} \int_{t=0}^x G^{(j)}(x - t - y) e^{-\lambda t} \frac{(\lambda t)^j}{j!} dW_3(y) dW(t - Q);$$

$$w_1^{(2)} = (w_1 + Q) / (1 - \rho_1) + w_1^{(3)};$$

$$T_t^*(s) = \Omega_3(s) \cdot \Omega(s + \lambda - \lambda\Gamma(s)) \cdot \text{EXP}[-t(s + \lambda - \lambda\Gamma(s))] \quad \text{if } t > Q;$$

$$T_{1/t} = (w_1 + t) / (1 - \rho_1) + w_1^{(3)} \quad \text{if } t > Q.$$

(For details see [11].)

Note II.2.: The expectation  $w_1^{(1)}$  should be  $w_1^{(1)} = \lambda(\alpha_2 - 2Q(\alpha - a)) / 2(1 - \lambda a)$  and the factor  $1/(1 + p)$  may not occur in this formula. (See Guimaraes, par. 4.)



Furthermore, Guimaraes gives a method to extend the foreground-background model to a model with  $K > 2$  queues in the case of the preemptive-resume priority rule and  $S = T = 0$ : Due to this priority rule, as far as the  $j$ -th queue is concerned ( $j = 1, \dots, K$ ), the queues  $j+1, \dots, K$  do not exist; also, any job in the  $j$ -th queue has to wait for the completion of all jobs in the queues  $1, \dots, j-1$ , plus for the service of jobs in front of the  $j$ -th queue, plus a delay caused by arrivals during these waiting times, plus a delay caused by arrivals during its own service in the  $j$ -th queue.

Note II.3.: Again, see note II.2, this last factor has been omitted in Guimaraes' work.

The order in which jobs are served in the queues  $1, \dots, j-1$  is irrelevant with respect to the job in the  $j$ -th queue. Constructing a modified process based on these considerations Guimaraes reduces the problem to the FB-system. For the non-preemptive case the analysis of models with  $K > 2$  queues (also called Feed-Back algorithms) leads to the analysis of a complex Markov chain. With regard to waiting times this has been carried out by Schrage [12] and by Kleinrock and Muntz [9] for various priority disciplines. H. Keuning [10] chooses another way to obtain the expectations of waiting times for models with  $K > 2$  queues, various priority rules and constant swap times. Using a method introduced by Cobham [2] he splits an arriving job into portions corresponding to the required quanta of processing time. However, he meets with difficulties deriving the moments of order higher than one.

### III. Preemptive-resume priority rule.

#### 1. Introduction.

In the foregoing chapter a constant swap time has been attached to the service time of jobs, when they enter the CPU from the BQ for the first time, providing for some additional activities. Using the preemptive-resume priority rule, jobs can be interrupted in the BQ. And also, additional - maybe few - activities have to be performed just before resumption of an interrupted job. Therefore, we extend the model by assuming that a constant swap time  $R$  will be attached to the service time of an interrupted job in the BQ. This swap should be performed just before resumption. Now, different policies are imaginable to interrupt and later on to resume the processing of a job in the BQ:

- 1) Upon arrival of a job in the empty FQ during the processing of a job from the BQ, an immediate preemption of this processing is generated. When the FQ is empty again, the interrupted processing is resumed at the point where it has been stopped ... .
  - a. after the constant swap time  $R$  has been performed.
  - b. after the constant swap time  $R$  has been performed, except in the case that the interruption occurred during the swap. Then the CPU starts processing of the real service immediately after a new, complete swap time  $R$  without completing the preempted swap - if no new arrivals in the FQ disturb this whole.
  - c. after the constant swap time  $R$  has been performed in the case that the real service of the job was preempted. Otherwise, in the case that the treatment of a swap was preempted, the interrupted processing is resumed immediately at the point where it has been stopped without attaching a new swap time  $R$ .
  
- 2) Upon arrival of a job in the empty FQ during the treatment of a swap time -  $T$  or  $R$  - the preemption is only just delayed, till this swap has been completed, and the arriving jobs during this delay have to wait in the FQ. Otherwise, an immediate preemption is generated. When the FQ is empty again, the interrupted job may be resumed at the point where stopped after, eventually, a constant swap time  $R$  is performed first.

The policy 1.a. seems to be the "simplest" one from theoretical point of view. However, this policy will not be used, or hardly, in practical implementations. For it contains the danger that the total service time of a job in the BQ only grows and grows. This is possible when the expected length of intervals of time in which an empty FQ is found, amounts less than R units of time.

Though it looks like the policy 1.a., the policy 1.b. seems applicable in simple implementations, for the activities during a swap time R may be dependent of the current system loading.

Also the policy 1.c. looks like the previous ones and it seems applicable in simple implementations, too. For in certain cases it may occur that a fixed amount of work must be done during each swap time R. Thus a second swap would be abundant.

From the practical point of view the policies from 2. are the most interesting ones. For by them implementations are embraced, in which the change-over activities may and can be executed in "mutual exclusion". From theoretical point of view these policies appear to be considerably more complicated than the policies from 1. Since in these cases waiting jobs can be present in the FQ not only by its unfinished predecessors but also by an unfinished swap-time processing.

From these considerations the great difference between the analysis of the theoretical model and the analysis of the practical use of such a model is obvious. It is the former which is applied in this paper, though the latter has inspired the whole study.

## 2. The foreground queue.

The waiting time  $\eta_1$  is not influenced by swap times R, since the BQ does not hinder the FQ at all. Hence, the foreground queue behaves like an M|G|1 queue with general service distribution function  $\phi_1(x)$ .

Hence (see Kleinrock [8] among others)

$$\Omega_1(s) = s(1 - \rho_1)/(s - \lambda + \lambda\phi_1^*(s)) , \quad \text{Re}(s) \geq 0$$

$$w_1^{(1)} = \lambda t_2 / 2(1 - \rho_1)$$

$$T_t^*(s) = \Omega_1(s) \cdot e^{-st} , \quad \text{Re}(s) \geq 0, \text{ if } t \leq Q + S$$

$$T_{1/t} = w_1^{(1)} + t \quad \text{if } t \leq Q + S .$$

3. The background queue: the general case of the policy 1.

The waiting time  $\eta_2$  of a job requiring more than  $Q + S$  units of processing time will be influenced and determined by the future pattern of arrivals. Namely, an arriving job A will not only preempt the service of a job already being in the BQ and being processed by the CPU at the arrival of A, but also it will attach an amount of time (at most R) to its processing time. Analyzing  $\eta_2$  we consider the stochastic processes

$U_1(t)$  = the unfinished work in FQ at time t.  
(= the remaining time required to empty the FQ of all jobs present at time t given that no new jobs arrive)

and

$U_2(t)$  = the unfinished work in the BQ at time t.

These functions are sometimes referred to as the "virtual" waiting time at time t since, for a FIFO system, it represents how long a (virtual) job would wait in queue if it entered at time t; however, this waiting-time interpretation is good only for FIFO disciplines, whereas the unfinished work interpretation applies for all disciplines.

Kleinrock [8] uses something like it in his book to analyse the busy period length and from this the waiting time in an  $M|G|1$  queue (par. 5.8 and 5.10). His methods in those paragraphs have inspired the following analysis.

Consider the policy 1.a. and let  $\tau_i$  be the arrival time of the i-th job. An arriving job A will increase  $U_1$  with an amount of  $\min\{Q + S, \varphi_A\}$  and it will increase  $U_2$  with  $\varphi_2 = \max\{\varphi_A - (Q + S), 0\}$ . Furthermore, if A interrupts the processing of a job in the BQ, generally it increases  $U_2$  with an amount of R.

An example of the functions  $U_1$  and  $U_2$  during one "global" busy period  $Y$  could be:

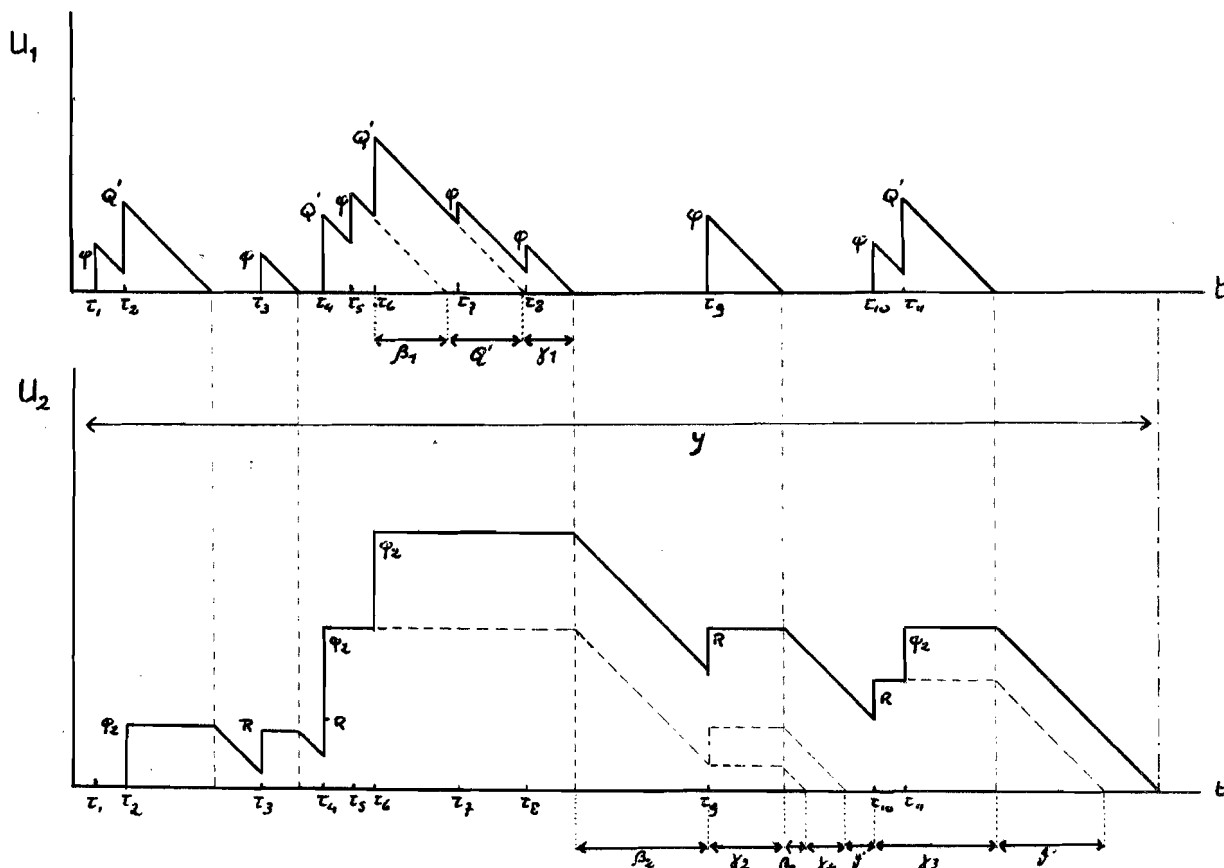


fig. III.1: The unfinished work  $U_1$  (in FQ) and  $U_2$  (in BQ) during  $Y$ .

The waiting time  $\eta_2$  of a job requiring more than  $Q + S$  processing time and arrived at time  $\tau_i$  can be split up, as follows:

- $\eta_2 = \beta_1$  : the unfinished work in the FQ at time  $\tau_i$ .
- +  $\beta_2$  : the unfinished work in the BQ at time  $\tau_i$ .
- +  $(Q + S)$  : the processing time of the job itself in the FQ.
- +  $\gamma_1$  : the delay caused by the jobs arriving during  $\beta_1 + (Q + S)$ .
- +  $\gamma_2$  : the delay in the FQ caused by the jobs arriving during the processing time being started with  $U(\tau_i) = \beta_2$ .
- +  $\gamma_3$  : the delay in the FQ caused by the jobs arriving during the processing time  $\theta$  (the attached swap times  $R$  included) of this job in the BQ.

+  $\gamma_4$  : the extra processing time caused by the swaps as a result of its own arrival and of the future jobs arriving during the processing time being started with  $U(\tau_i) = \beta_2$ .

In fig. III.1 the job arriving at time  $\tau_6$  requires more than  $Q + S$  processing time. The above components of  $\eta_2$  of this job have been pictured there. For the job arriving at time  $\tau_4$  we have  $\beta_1 = 0$  and  $\gamma_4$  contains the time needed to process the swap time  $R$  caused by its own arrival. And, finally,  $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0$  in the waiting time of the job arriving at time  $\tau_{11}$ .

Note that each job arriving during  $\beta_1 + (Q + S)$  may be viewed as initiating a local busy period in the FQ; each arriving job preempting the BQ processing initiates a local busy period in the FQ and enlarges the background processing with size  $R$ ; the local busy periods of  $\gamma_1$ ,  $\gamma_2$  and  $\gamma_4$  do not depend on each other.

Now, let  $X_i$  be the length of the  $i$ -th foreground busy period and  $\Gamma$  its LST from the  $M|G|1$  system with service distribution function  $\phi_1$ .

By use of the law of total probability and by conditioning on

$$(\beta_1 + (Q + S) = x; \beta_2 + \gamma_4 = u; \theta = y; j \text{ arrivals in } x + y + u)$$

it follows that

$$\begin{aligned} \Omega_2(s) = & \Delta^*(s + \lambda - \lambda\Gamma(s)) \times \theta^*(\lambda - \lambda\Gamma(s)) \times \\ & \times B_1^*(s + \lambda - \lambda\Gamma(s)) \times \text{EXP}[-(Q + S)(s + \lambda - \lambda\Gamma(s))] , \end{aligned}$$

where  $\Delta^*$ ,  $\theta^*$  and  $B_1^*$  are the LST of the distribution function of  $\delta = \beta_2 + \gamma_4$ ,  $\theta$  and  $\beta_1$  respectively and  $\text{Re}(s) \geq 0$ . For a detailed proof of this formula see [11].

Since the FQ may be viewed upon as an  $M|G|1$  queue, it follows (see Kleinrock [8] among others) that

$$B_1^*(s) = s(1 - \rho_1)/(s - \lambda + \lambda\phi_1^*(s)) , \quad \text{Re}(s) \geq 0 .$$

The LST's  $\theta^*$  and  $\Delta^*$  will depend on the policy used, in as much as this policy determines the amount of swap time, which will be attached to the background-

processing time.

Finally it is easy to see that for  $t > Q + S$

$$T_t^*(s) = \Omega_2(s) \cdot \text{EXP}[-s(t - (Q + S))] , \quad \text{Re}(s) \geq 0 .$$

4. The background queue:  $\theta^*$  and  $\Delta^*$  for the policy 1.a.

Let's consider the workload of a job A in the BQ throughout its processing time  $\theta$ . Only the arriving jobs preempting this processing influence the time  $\theta$  by attaching a time  $R$  to it. Let  $T'$  be the relative time in which the CPU processes job A. Then the workload is of the form as pictured in fig. III.2.

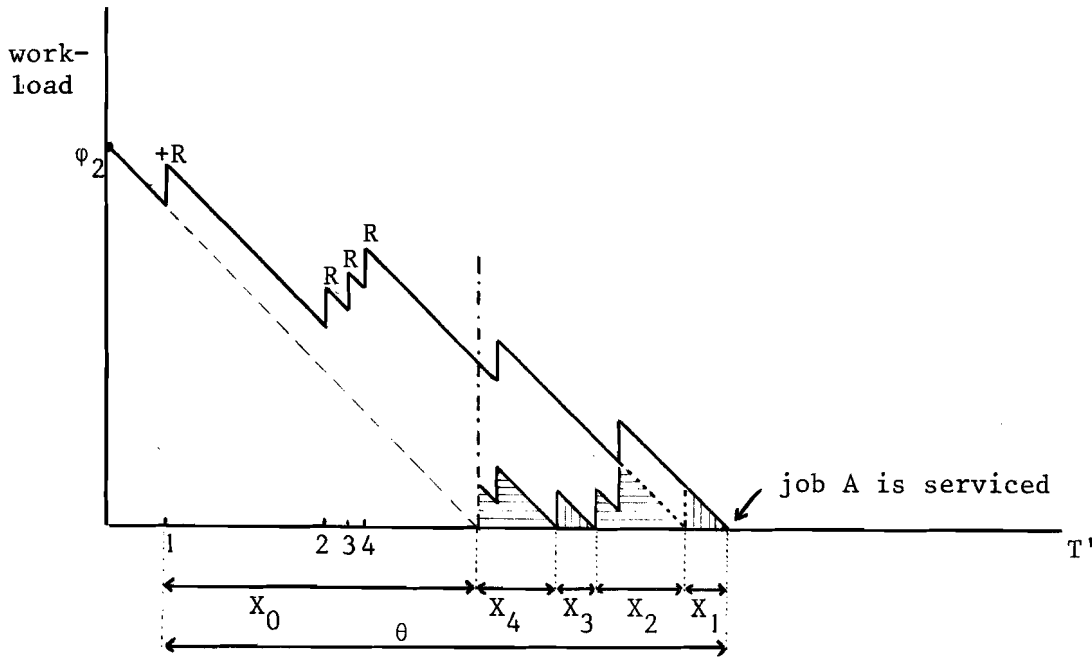


fig. III.2: the amount of work of a job in the BQ.

This function may be viewed upon as the unfinished work for one job. Hence, this function is independent of the order of service! Therefore we may divide the time  $\theta$  in parts of length  $X_i$ , where  $X_0$  equals the length  $\phi_2$  and  $X_i$  equals the duration of the completion of the  $i$ -th swap being attached during  $X_0$ , including the attached swap times  $R$  during this completion itself. Defining  $v$  as the number of preemptions during  $X_0$  it holds  $0 \leq i \leq v$  and

$$\theta = X_0 + \sum_{i=1}^v X_i, \text{ where the lengths } X_i, i > 0, \text{ are independent and identically}$$

distributed with distribution function  $\Sigma$ ; the LST is  $\Sigma^*$ . The length  $X_0$  is independent of the  $X_i$ 's and it has distribution function  $\phi_2$  obviously. By conditioning on  $(X_0 = x; v = j)$  and by the law of total probability it follows - compare [11] - that

$$\theta^*(s) = \phi_2^*(s + \lambda - \lambda \Sigma^*(s)), \quad \text{Re}(s) \geq 0 .$$

Applying the same method on  $X_i - i > 0$  - instead of on  $\theta$  and defining the LST of the Dirac delta probability density function  $u_0(x - R)$  by  $R^*$  it follows that

$$\Sigma^*(s) = R^*(s + \lambda - \lambda \Sigma^*(s)) = \text{EXP}[-R(s + \lambda - \lambda \Sigma^*(s))] \quad , \quad \text{Re}(s) \geq 0$$

and it is easy to show that this equation in  $\Sigma^*$  has one unique solution within the class of LST's of distribution functions.

To derive  $\Delta^*$ , firstly we need  $Z^*$ , the LST of the background busy period length distribution. Using a method firstly published by Conway e.a. [4] and used by Kleinrock [8, par. 5.10] we get  $B_2^*$  - the LST of the distribution function of  $\beta_2$  - and after that  $\Delta^*$ .

So let's consider the unfinished work  $U_2$  during one background busy period measured in time ignoring the time spent on the FQ. An example based on fig. III.1 is given in fig. III.3.

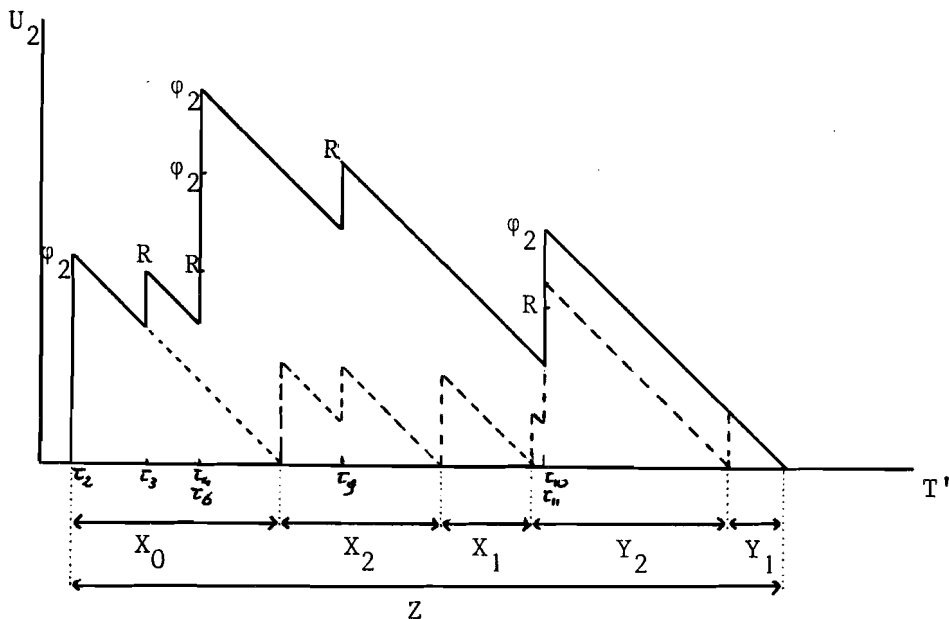


fig. III.3: the unfinished work  $U_2$  in BQ during one background busy period; LCFS-discipline.



The background busy period length  $Z$  is independent of the order of service! Therefore we may divide  $Z$  in a sequence of intervals with length  $X_i$  and  $Y_j$  ( $i \geq 0, j > 0$ ). The duration of the first interval  $X_0$  equals  $\phi_2$ , which is the total remaining processing time of the job initiating this busy period. For  $i > 0, X_i$  is the length of time required to process the  $i$ -th job which joins the BQ during  $X_0$ , and all those jobs and swaps which occur during this processing of the  $i$ -th job. For  $j > 0, Y_j$  is the length of time required to process the  $j$ -th swap which has been attached during  $X_0$ , and all those jobs and swaps which occur during this processing of the  $j$ -th swap.

Define  $v$  as the number of jobs with  $\phi > Q+S$  joining the BQ during  $X_0$  and  $w$  as the number of jobs attaching R on  $U_2$  during  $X_0$ . The order of service is assumed to be:  $X_0, X_v, \dots, X_1, Y_w, \dots, Y_1$ . Obviously it holds that

$$Z = X_0 + \sum_{i=1}^v X_i + \sum_{i=1}^w Y_i, \text{ the interval lengths } X_i \text{ (} i \geq 0 \text{) and } Y_j \text{ (} j > 0 \text{)}$$

do not depend on each other,  $X_0$  has distribution function  $\phi_2$  and the  $X_i$ 's ( $i > 0$ ) are distributed exactly the same as  $Z$ . Therefore  $Z^*(s) = \epsilon e^{-sX_i}$  ( $i > 0$ ), too!

Defining  $T^*(s) = \epsilon e^{-sY_i}$  for  $i > 0$  and  $\text{Re}(s) \geq 0$  it holds that

$$\epsilon(e^{-sZ} \mid X_0 = x, v = j, w = k) = e^{-sx} [Z^*(s)]^j [T^*(s)]^k .$$

To remove these conditions we need the following probability distributions

1.  $\Pr(X_0 \leq x) = \phi_2(x)$
2.  $\Pr(w = k \mid X_0 = x) = e^{-\lambda x} (\lambda x)^k / k! =: P_k(x)$   
 viz. only the jobs which arrive within  $X_0$  during the very processing of the first job, preempt this processing and attach R to  $U_2$
3.  $\Pr(v = j \mid X_0 = x, w = k)$

Following Guimaraes "a busy period initiated by  $k$  jobs" is a busy period, in which  $k$  jobs have been or appear to have been arrived in front of it; and  $f_j^{(k)}$  is the probability of  $j$  arrivals in the BQ during a busy period initiated by  $k$  jobs.

Now,  $K$  preemptions occur during  $X_0 = x$  with the probability  $\Pr(w = k \mid X_0 = x)$  and each of them initiates a foreground busy period. The probability of  $j$  arrivals in the BQ during these  $K$  foreground busy periods equals the probabi-

lity of  $j$  arrivals in the BQ during a foreground busy period initiated by  $K$  jobs.

Hence  $\Pr(v = j \mid X_0 = x, w = k) = f_j^{(k)}$ .

A busy period length initiated by  $K$  jobs in the FQ is equally distributed as the sum of  $K$  independent busy periods initiated by 1 job in the FQ - therefore as the  $K$ -th convolution of the density function of  $f_j^{(1)}$ . Let  $F^{(k)}(w)$  be the transform of  $f_j^{(k)}$  and  $F(w) = F^{(1)}(w)$ . Then (see Guimaraes)

$$f_j^{(1)} = \sum_{k=0}^{\infty} \left\{ \int_0^{Q+S} P_k(t) \cdot d\phi(t) \cdot f_j^{(k)} + p \cdot P_k(Q+S) \cdot f_{j-1}^{(k)} \right\},$$

$F^{(k)}(w) = [F(w)]^k$  and  $F(w)$  is the unique solution of

$$F(w) = \int_0^{Q+S} \text{EXP}[-\lambda t(1 - F(w))] d\phi(t) + p \cdot w \cdot \text{EXP}[-\lambda(Q+S)(1 - F(w))].$$

Now, we are able to remove the three conditions and as proved in [11] it holds that

$$Z^*(s) = \phi_2^*(s + \lambda - \lambda T^*(s) \cdot F[Z^*(s)]), \quad \text{Re}(s) \geq 0.$$

To derive  $T^*$  we may apply the same method on  $Y_i, -1 \leq i \leq w$ , as we used to derive  $Z^*$ . Splitting up  $Y_i$  in a sequence of intervals as we did for  $Z$  and by the fact that each processing of  $Y_i$  starts with a fixed amount of work, viz.  $R$ , it holds that

$$T^*(s) = \text{EXP}[-R(s + \lambda - \lambda T^*(s) \cdot F[Z^*(s)])], \quad \text{Re}(s) \geq 0.$$

Taking the first derivative of  $Z^*$  and setting  $s = 0$  it follows after a straightforward computation that the average background busy period length is

$$eZ = \frac{u_1}{1-\lambda R} \cdot \frac{1}{1 - \frac{\rho_2}{1-\rho_1}} = \frac{u_1}{1-\lambda R} \cdot \frac{1-\rho_1}{1-\rho_{\text{tot}}},$$

where  $\rho_{\text{tot}} = \rho_1 + \rho_2$ ;  $\rho_1, \rho_2$  have been defined earlier as the utilization factor in the FQ and BQ respectively.

Note III.1 : By definition  $\rho_1 = \lambda t_1$  and

$$\rho_2 = \lambda P\left(-\frac{d}{ds} \theta^*(s) \mid s = 0\right) = \lambda p u_1 / (1 - \lambda R) .$$

Note III.2 : An intuitive derivation and interpretation of this simple result can be found in [11]. The LST  $Y^*$  belonging to the global busy period length and its expectation is derived there, too.

Since  $\beta_2$  equals the unfinished work  $U_2$  at an arrival moment, the order of service plays an important part in the derivation of the LST  $B_2^*$ , and this order may no longer be chosen freely.

Following Kleinrock, now, we divide one background busy period length  $Z$  in a sequence of intervals whose lengths  $X_i$  ( $i \geq 0$ ) are dependent random variables. An example based on fig. III.1 is given in fig. III.4.

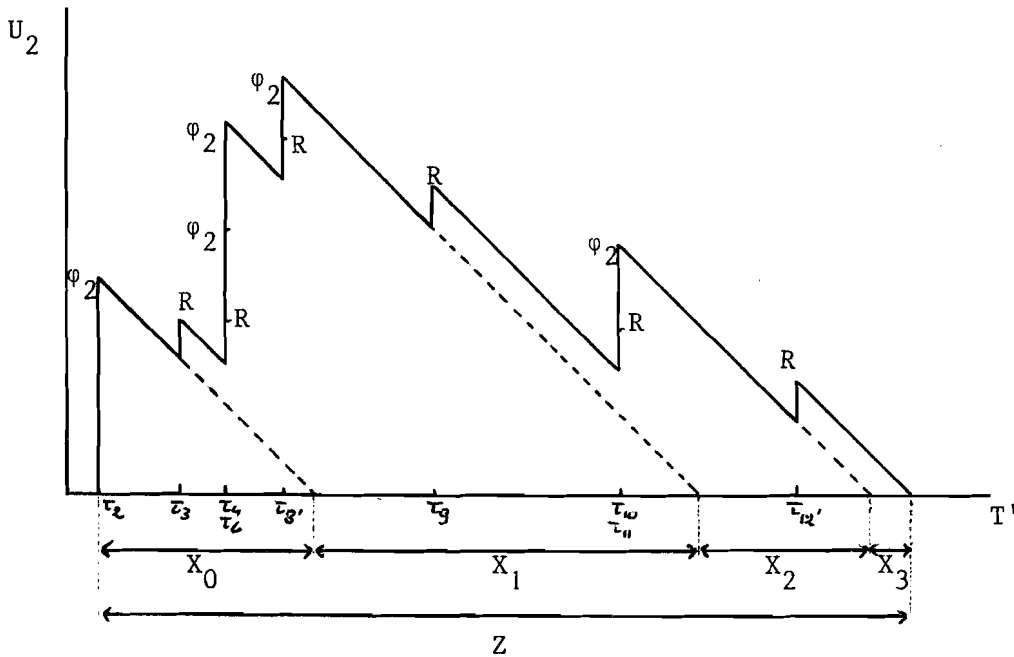


fig. III.4: The unfinished work  $U_2$  in BQ during one background busy period; FCFS-discipline.

The duration of the first interval  $X_0$  equals  $\varphi_2$ , which is the total remaining processing time of the job initiating this busy period. For  $i > 0$ ,  $X_i$  is the length of time required to process all those jobs and swap times  $R$ , which

arrive in the BQ or have been attached during the previous interval whose duration is  $X_{i-1}$ . We let  $v_i$  equal the number of jobs which arrive in the BQ during  $X_i$  and  $w_i$  equals the number of swap times R which have been attached during  $X_i, i \geq 0$ . Thus we see that

$$Z = \sum_{i=0}^{\infty} X_i,$$

where the possibility of an infinite sequence of such intervals is permitted. Clearly, we define  $X_i = 0$  for those intervals that fall beyond the termination of this busy period; for  $\rho < 1$  it is known that with possibility 1 there will be a finite  $i_0$  for which  $X_{i_0}$  and all its successors will be 0. Furthermore, we know that  $X_{i+1}$  will be the sum of  $v_i$  background-processing intervals -each of which distributed as  $\phi_2$ - plus the sum of  $w_i$  swap-time intervals -each of which with density  $u_0(x - R)$ . Defining  $X_i(y) = \Pr(X_i \leq y)$  and  $X_i^*$  its LST we are able to derive a recurrence relation among the  $X_i^*$  -see [11]-

$$X_i^*(s) = X_{i-1}^*(\lambda - \lambda R^*(s) \cdot F[\phi_2^*(s)]), \quad i \geq 1 \text{ and } \text{Re}(s) \geq 0.$$

Now, we consider the time  $\beta_2$  of a tagged job A which arrives in the BQ during Z and, in particular, in the interval of duration  $X_i$ . By renewal theory arguments it can be made rigorous that the probability of finding the system in this state  $-X_i-$  for Poisson arrivals equals  $\epsilon X_i / \epsilon Z$ .

Clearly, for this job A the time  $\beta_2$  equals the sum of (1) the remaining time of  $X_i$  (i.e. the residual life  $Y_i$ ), (2) the swap times which have been attached during  $X_i$  already before the arrival of A in the BQ (the number equals  $W_i$  in duration  $X_i - Y_i$ ) and (3) the background-processing intervals of all jobs, which arrived during  $X_i$  already before the arrival of A in the BQ (the number equals  $V_i$  in duration  $X_i - Y_i$ ).

Thus, by conditioning on  $(X_i = x; Y_i = y; W_i = k; V_i = j)$ , by use of the joint probability density function of  $X_i$  and  $Y_i$  -which is known from renewal theory-, by use of the recurrence relation among the  $X_i^*$  and by the knowledge that  $X_{i_0+j}^* = 1$  ( $j \geq 0$ ) and  $X_0^* = \phi_2^*$  it follows

$$\begin{aligned} & \epsilon(e^{-s\beta_2} \mid \text{enter in busy period}) \\ &= \sum_{i=0}^{\infty} \epsilon(e^{-s\beta_2} \mid \text{enter in } X_i) \cdot \epsilon X_i / \epsilon Z \end{aligned}$$

$$\begin{aligned}
 &= \frac{\sum_{i=0}^{\infty} [X_{i+1}^*(s) - X_i^*(s)]}{[s - \lambda + \lambda R^*(s)F(\Phi_2^*(s))] \epsilon Z} \\
 &= \frac{1 - \Phi_2^*(s)}{[s - \lambda + \lambda R^*(s)F(\Phi_2^*(s))] \epsilon Z}
 \end{aligned}$$

Since  $\rho_2$  equals the probability to arrive in a background busy period we get

$$B_2^*(s) = 1 - \rho_2 + \lambda p \frac{1 - \rho_{\text{tot}}}{1 - \rho_1} \cdot \frac{1 - \Phi_2^*(s)}{[s - \lambda + \lambda R^*(s)F(\Phi_2^*(s))]} , \quad \text{Re}(s) \geq 0 .$$

Since for each job A  $\gamma_4$  equals the extra processing time caused by the swaps as a result of its own arrival and of the future arrivals of jobs arriving during the processing time of A being started with  $U_2(\tau_i) = \beta_2$  we define

$w = \begin{cases} 0 & \text{if the background-processing is not interrupted by the arrival of A} \\ 1 & \text{if the background-processing is interrupted by the arrival of A.} \end{cases}$

Then it holds that  $\text{Pr}(w = 1) = \rho_2$ , because job A interrupts the background-processing only when it arrives during a background busy period. Again, the unfinished work function  $U_2$  does not depend on the order of service during an interval of duration  $\delta$ . Therefore, by conditioning on  $w$  it is to show that (compare [11])

$$\Delta^*(s) = B_2^*(s + \lambda - \lambda \Sigma^*(s)) \cdot \{1 - \rho_2 + \rho_2 \Sigma^*(s)\} , \quad \text{Re}(s) \geq 0$$

and where the LST  $\Sigma^*$  has been defined as in the derivation of  $\theta^*$ .

Concluding  $\Omega_2$ , the LST of the distribution function of the total waiting time in system ( $\eta_2$ ), is known and also the  $r$ -th moments of it could be derived, now. Furthermore, it holds that for  $t > Q + S$

$$T_t^*(s) = \Omega_2(s) \cdot \text{EXP}[-s(t - (Q + S))] .$$

5. The background queue:  $\theta^*$  and  $\Delta^*$  for the policy 1.b.

Comparing the policy 1.b with the policy 1.a we see that the only difference between them occurs in the amount of work which will be attached to the function  $U_2$  by new arrivals. However, using the policy 1.b the attached swap time caused by a preemption of the background-processing, while only  $x < R$  units of time are spent since the previous preemption, amounts exactly  $x$ . In the other case it amounts  $R$ .

Let  $r$  be the amount of work attached to  $U_2$  caused by a preemption of the background-processing and  $r(x)$  its probability density function. Using policy 1.a  $r(x) = u_0(x - R)$ , the Dirac delta probability density function. Now, using the policy 1.b and because of the Poisson-arrival process with rate  $\lambda$  it holds that  $\Pr(x \leq r < x + dx \text{ and preemption during swap-time processing})$

$$= \begin{cases} 0 & \text{if } x \geq R \\ \Pr(x \leq \text{interarrival time} < x + dx) = \lambda e^{-\lambda x} & , \text{ if } 0 \leq x < R \end{cases}$$

and

$$\begin{aligned} & \Pr(x \leq r < x + dx \text{ and preemption not during swap-time processing}) \\ &= u_0(x - R) \cdot \Pr(\text{preemption not during swap-time processing}) \\ &= u_0(x - R) \Pr(\text{interarrival time} \geq R) \\ &= u_0(x - R) e^{-\lambda R}. \end{aligned}$$

Hence

$$r(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } 0 \leq x < R \\ u_0(x - R) e^{-\lambda x} & , \text{ if } x \geq R \end{cases}$$

and

$$R^*(s) = \frac{1}{s+\lambda} [\lambda + s \cdot \text{EXP}[-R(s + \lambda)]], \quad \text{Re}(s) \geq 0$$

and

$$er = [1 - e^{-\lambda R}] / \lambda .$$

Using this  $R^*$ , in exactly the same manner as in III.4 the LST  $\Omega_2$  can be derived for the policy 1.b.

6. The background-queue:  $\theta^*$  and  $\Delta^*$  for the policy 1.c.

Using the policy 1.c it holds that

$$\begin{aligned} & \Pr(x \leq r < x + dx \text{ and preemption during swap-time processing}) \\ &= u_0(x) \cdot (1 - e^{-\lambda R}) \text{ and} \end{aligned}$$

$$\begin{aligned} & \Pr(x \leq r < x + dx \text{ and preemption not during swap-time processing}) \\ &= u_0(x - R) e^{-\lambda R} . \end{aligned}$$

$$\begin{aligned} \text{Hence } r(x) &= u_0(x)(1 - e^{-\lambda R}) + u_0(x - R)e^{-\lambda R}, \\ R^*(s) &= 1 - e^{-\lambda R} + \text{EXP}[-R(s + \lambda)], \quad \text{Re}(s) \geq 0 \\ \varepsilon r &= \text{Re}^{-\lambda R}. \end{aligned}$$

Now, using this  $R^*$ , in exactly the same manner as in III.4 the LST  $\Omega_2$  can be derived for the policy l.c.

### 7. Extension to models with $K > 2$ queues.

Like Guimaraes, see chapter II, we could try to extend our model to models with  $K > 2$  queues using the preemptive-resume priority rule and attaching swap times before resumption in each queue. This could be done because of the same considerations as used by Guimaraes. Namely, due to the priority rule, as far as the  $j$ -th queue is concerned ( $j = 1, \dots, K$ ), the queues  $j+1, \dots, K$  do not exist; the order of processing the jobs in the queues  $1, \dots, j-1$  is irrelevant for the waiting time  $\eta_2^1$  of a job in the  $j$ -th queue; also now, a modified process could be defined, in which the jobs in the queues  $1, \dots, j-1$  could be processed in an order which may be chosen freely, yet. And so we could try to reduce the problems to the FB-system splitting  $\eta_2^1$  as done for  $K = 2$ . However, a complete analysis of the problems met and of their solutions is not available, till now. The amount of work, which has to be performed by the CPU for a certain job, is influenced by the pattern of future arrivals and so the independence among the jobs disappear.

### 8. The policy 2: delayed preemptive-resume priority.

Only a few remarks will be mentioned in this paper about the policy 2 as described in III.1. Firstly, the case  $R = 0$  -hence a delay of preemptions may occur only during swap time  $T$ - should be studied for getting experience with this policy. One of the first tasks should contain an analysis of the duration of the foreground busy period and the processing-time distribution in the FQ. Doing this it could be helpful to split the stochastic process "unfinished work in system" in (1) the unfinished work in the FQ, (2) the unfinished work concerning  $T$ -swap times and (3) the remaining work in the BQ. In those analyses, in which the order of service is insignificant, the first two processes could be considered as an FB-system with non-preemptive priority rule and in which the background-processing contains only the  $T$ -swap times. Then the results of Guimaraes are available.

Yet, an analysis of the unfinished work seems useful, too, in the cases in which the order of service plays an important part. Thus, an eventually adapted method as used in the derivation of  $B_2^*$  and some use of the renewal theory appears to be useful, here too. After analysing the above case  $R = 0$  it does not appear to be very difficult to extend the model for  $R \neq 0$ . From practical point of view it looks interesting in any case.



References.

- [ 1] I. Adiri and Avi-Itzhak, A time-sharing queue.  
Management Science 15 (1969), 639-657.
- [ 2] A. Cobham, Priority assignment in waiting line problems.  
Operation Research 2 (1954), 70-76.
- [ 3] E.G. Coffman, Markov chain analysis of multiprogramming computer systems.  
Naval Res. Log. Quart. 16, 2 (1969), 175-197.
- [ 4] R.W. Conway, W.L. Maxwell and L.W. Miller, Theory of scheduling.  
Addison-Wesley 1967.
- [ 5] D.W. Fife, An optimization model for time-sharing.  
Proc. AFIPS 66, SJCC, vol. 28, 97-104.
- [ 6] C. Guimaraes, A foreground-background timesharing queue with general service times.  
Lecture notes in Comp. Science 16,  
Springer-Verlag, Berlin 1974, 103-125.
- [ 7] L. Kleinrock, Analysis of a time-shared processor.  
Naval Res. Log. Quart. 2, 1 (1964), 59-73.
- [ 8] L. Kleinrock, Queueing systems, vol. I: theory.  
John Wiley and Sons, New York 1975.
- [ 9] L. Kleinrock and Muntz, Processor sharing queueing models of mixed scheduling disciplines.  
JACM 19 (1972), 453-481.
- [10] H. Keuning, A queueing system with priority scheduling.  
Master's thesis, Department of Mathematics, Eindhoven University of Technology 1974.
- [11] J. van de Schoot, Een wachtvrij model voor een foreground-background time-sharing systeem met algemene bedieningstijd.  
COSOR-research note R-76-12. Department of Mathematics, Eindhoven University of Technology 1976.

- [12] L.E. Schrage, The queue  $M|G|1$  with feedback priority queue.  
Management Science 13 (1967), 466-474.