# Stationary Load Conditions and Virtual Queues in a multiprocessor networks.


## N. Andronaty, Dm. Mishkoy, Gh. Mishkoy and I. Andronaty

Institute of Mathematics and Informatics of the Academy of Science of Moldova, Free International University of Moldova

andronat@cc.acad
gmiscoi@ulim.moldnet.md
str. Academiei 5, MD-2080, Kishinev, Moldova



**Keywords:** *switching, priority, distribution, queue*



**Abstract:** *The work is dedicated to study the processing of the non homogeneous tasks in a multiprocessor network, the tasks being previously divided into classes of urgency. Load conditions , virtual and steady state distribution of queues length are obtained.*


## Preface

When organizing parallel computations in transputer networks much attention is paid to load control algorithms. Methods and tools of queueing theory are successfully used for this purpose as effective means to model and study these algorithms. Stochastic nature of the processes, of their handling, and of interaction of information flows, which are, as a rule, inhomogeneous, can result in overload of some parts or nodes of the network, whereas another parts of the system may be idle. The approach based on the queuing theory was successfully used by a number of authors to analyze process evolution in complex systems such as information and computation ones (see, for example, [1]). The remarkable examples of such analysis are work [2], investigating a class of dynamic load balancing algorithms which uses probabilities of task transference in a M|M|1 queuing system, and article [3] which uses M|G|1 queuing system to examine scheduling procedures in a single-processor real-time computer system.

Note, that both in articles mentioned above, and in a number of others, requests in the input flow are assumed homogeneous. It seems to us that in many cases heterogeneity (preemptive character) of the request flow is more natural and that it describes more adequately actual processes taking place in a multiprocessor network. In some other cases, introducing various preemptive service strategies, one can succeed in redistribution of arriving tasks, which results in raising the efficiency of computing process and in optimization of some its parameters.

## Mathematical model

We assume, that the input task flow divides into an arbitrary number r, of homogeneous task classes. The service time $B_i$ by a host i, i=1,...,r, is assumed to be a random variable with distribution function $B_i(x)$ of an arbitrary kind. Homogeneous request classes obtained as a result of division of the original request flow are called below urgency classes or priority classes. A value meaning the level of urgency or the priority is assigned to each class. We assume that the classes are enumerated in a descending order of their priorities. To find the order in which requests belonging to different classes are chosen for service, some priority rules are then defined , they establish a measure of preemption of requests belonging to one class with respect to requests of another one. The order of choosing the requests belonging to the same class is defined also. For more details about the question see, for example, monographs [4], [5]. This queuing system is designated as . In the case r=1 (a single homogeneous input flow or a single priority class), the queuing system would be, naturally, of M|G|1 kind. The more particular case, when G=M, i.e. it is known that the service time is distributed exponentially, the queuing system would be M|M|1.

Let us return to the $M_r|G_r|1$ queuing system and make some additional assumptions. We assume that at each node there exist one another stochastic process besides the process of service of arriving requests. This additional stochastic process models loss in time for so-called switching, or changing the orientation to the respective priority class. For example, there can exist losses in time to transfer and analyses the information amongst load balancing programs, while message routing in the network is performed. We assume that this losses are random variables designated , having their distribution functions $C_{ij}(x)$, i,j=1,...,r; $i \neq j$. Indices i, j mean switching of the processor to the tasks of j priority class, provided that the previous task was from i class. The condition $i \neq j$ means that time losses to switch inside the same priority class are equal to zero. That is, time losses exist only when the priority class of requests changes. To simplify mathematical notation in the text below, in the case $r \geq 2$ we assume that distribution functions depend only on index j. That is, we assume, that random variables $C_{ij} = C_j$, and, respectively, their distribution functions $C_{ij}(x) = C_j(x)$. It means, that the time loss to switch the processor from i priority class to j one depends only on j.

We assume also that as soon as the period of activity of the processor (its busy state time interval) completes, its current switch state (its orientation) is immediately lost. The reset operation occurs instantly. At the beginning of the next period of activity the service starts only after random time interval $C_j^0$, depending only on the priority class index j, j=1,...,r. This value may not be obligatory equal to $C_j$. However, although the general case can be examined, we assume that $C_j^0 = C_j \neq 0$, thus slightly simplifying the general model. Note, that there exist no restrictions about both the kind of distribution functions $C_{ij}(x)$, and the kind of distribution functions $B_i(x)$. This permits us to use described above $M_r|G_r|1$ queuing system in a wide range of modeling.

## Stationary load conditions

Before considering the main results of the work let us analyses two particular classes of the general $M_r|G_r|1$ queuing system with time losses for switching the priority classes. Sections 3.1 and 2.2 consider the homogeneous case (without priorities). Namely, in section 2.1

the case of r=1, $C_1 = C_1^0 = 0$ is considered, while in section 2.2 the case of r=1, $C_1^0 \neq 0,\ C_1 = 0$ is considered.

## Elementary homogeneous case

Let us assume that the input task flow is a Poisson one with the parameter a. Task service time is assumed to be a random variable with its distribution function B(x). Let us also denote the distribution function of the length of busy period $\prod$ by $\prod(x) = P\{\prod \le x\}$ , and by $\pi(s)$ and $\beta(s)$ Laplace-Stieltjes transforms of the functions B(x) and $\prod$ (x) respectively, i.e.

$$\pi(s) = \int_0^\infty e^{-st} d\prod(t), \quad \beta(s) = \int_0^\infty e^{-st} dB(t),$$

by $\beta_1$ and $\pi_1$ first moments of this functions.

It is known, that the following functional equation

$$\pi(s) = \beta(s + a - a\pi(s)) \quad (2.1)$$

determines the unique function $\pi(s)$ being Laplace-Stieltjes transform of distribution function $\pi(t)$ , which is analytical when

Distribution function $\prod$ (t) is a proper one when $a\beta_1 \le 1$ and an improper one when $a\beta_1 > 1$.

Moreover, if

$\rho = a\beta_1 < 1,$ (2.2)

then $\pi_1 < \infty$, and

$$\pi_1 = \frac{\beta_1}{1 - a\beta_1} \quad (2.3)$$

Thus, in the case condition (2.2) does not hold, the queue length grows infinitely, and the processor is incapable to serve all the arriving requests.

## Homogeneous case with non-zero losses.

The difference between the queuing system used in this section and the previous one is that each time before busy period starts and the processor is ready to begin the service, an additional segment of time of random length $C_1^0$ must expire. Its distribution function we denote by $\Gamma(x) = P\{C_0^1 \le x\}$, while Laplace-Stieltjes transform of this function by $\gamma(s)$. Let's assume

$$\gamma_1 = \int_0^\infty t d\Gamma(t) < \infty.$$

Now let us present formulas analogous to the (2.1) to (2.3) from the previous section using the same notation. Through $\prod^R(t)$ we denote the distribution function of the busy period length, while through $\pi^R(s)$ its Laplace-Stieltjes transform. It can be proved, that

$\pi^R(s) = \gamma(s + a - a\pi(s))\pi(s),$ (2.4)

$\pi(s) = \beta(s + a - a\pi(s))$

If

$\rho = a\beta_1 < 1,$ (2.5)

then

$$\pi_1^R = \frac{\gamma_1 + \beta_1}{1 - a\beta_1}, \quad (2.6)$$

where $\pi_1^R$ is the average busy period length.

Let's note, that condition (2.5) is the same as condition (2.2) for the previous queuing system.

## General multidimensional case.

In this section a general basic algorithm is described. It deals with a multidimensional (multi-class) task flow, arbitrary service time distribution functions for each priority class, and arbitrary switching time losses distribution functions for each class. The basis of this algorithm is a multidimensional analogue of formula (3.2), deduced as a result of study of a general $M_r | G_r | 1$ queuing system with time losses for task class switching.

Let us denote by $a_i$ the parameter of an i-th input Poisson flow, where i=1,2,...,r, $r \ge 2$ ; $B_i(x)$ let be the distribution function of service time length for requests from i-th priority class, and $C_j(x)$ be the distribution function of time losses for switching the calculation process to the class j, j=1,...,r, $i \neq j$. Input flows corresponding to the priority classes are assumed to be enumerated in a descending order of their priorities. The priority is assumed to be absolute as it concerns both service process and switching. This means the following: if during the service of a particular task another task with a greater priority arrives, then the service process is interrupted; just the same if during the switching period to the particular task another task with a greater priority arrives, the process of switching is interrupted. Various strategies about how the interrupted processes are managed afterwards could be considered.

Let's introduce some more notations: let $\prod_k(x)$ be the distribution function of busy period length for all the priorities including the k-th one and the greater ones, and $\prod(x)$ be the distribution function of busy period length. Note [4], that $\prod_r(x) = \prod(x)$. Furthermore, let's assume that

$$\sigma_k = a_1 + ... + a_k, \quad \sigma = \sigma_r, \quad \sigma_0 = 0;$$

$$\beta_i(s) = \int_0^\infty e^{-sx} dB_i(x), \quad c_j(s), \quad \pi(s), \quad \pi_k(s)$$

are Laplace-Stieltjes transforms of distribution functions $B_j(x),\ C_j(x),\ \prod(x),\ \prod_k(x)$ respectively $\beta_{k1},\ c_{k1},\ \pi_{k1},...$ are first moments of functions $B_k(x),\ C_k(x),\ \prod(x),\ \prod_k(x),...,k = 1,...,r$ respectively.

The condition of stationary load for the processor is

$$\rho_k = \sum_{i=1}^k a_i b_i < 1, \quad (2.7)$$

where

$$b_1 = \frac{\beta_{11} + c_{11}}{1 + a_1 c_{11}}$$

$$b_i = \Phi_1 ... \Phi_{i-1} \frac{1}{\sigma_{i-1}} [\frac{1}{\beta_i \sigma_{i-1}} - 1](1 + \sigma_{i-1} c_{i1}),$$

$$\Phi_1 = 1, \quad \Phi_i = 1 + [\sigma_i - \sigma_{i-1}\pi_{i-1}(a_i)]\, c_{i1}, \quad i = 2,...,k.$$

Functions $\pi_{i-1}(a_i)$ are found using the following system of recurrent equations:

$$\sigma_k \pi_k(s) = \sigma_{k-1}\pi_{k-1}(s + a_k) + \sigma_{k-1}\{\pi_{k-1}(s + a_k[1 - \bar{\pi}_k(s)]) -$$
$$- \pi_{k-1}(s + a_k)\}\upsilon_k(s + a_k[1 - \bar{\pi}_k(s)]) + a_k \pi_{kk}(s), \quad (2.8)$$

$$\pi_{kk}(s) = \upsilon_k(s + a_k[1 - \bar{\pi}_k(s)])\bar{\pi}_k(s),$$

$$\bar{\pi}_k(s) = h_k(s + a_k[1 - \bar{\pi}_k(s)]),$$

where

$$h_k(s) = \beta_k(s + \sigma_{k-1})\{1 - \frac{\sigma_{k-1}}{s + \sigma_{k-1}}[1 - \beta_k(s + \sigma_{k-1})]\pi_{k-1}(s)\upsilon_k(s)\}^{-1}, \quad (2.9)$$

$$\upsilon_k(s) = c_k(s + \sigma[1 - \pi_{k-1}(s)]). \quad (2.10)$$

Here for purposes of definiteness it is assumed, that an interrupted (in consequence of a task with a greater priority arrival) service of a particular task is restarted, while interrupted switching starts form the interruption point.

In the case condition (2.7) holds, we can find from equation (2.10) to (2.12) average values not only for the busy period length, but for a number of other functions.

Namely, if $\rho_k < 1$

$$\sigma_k \pi_{k1} = \frac{\Phi_2 ... \Phi_k + \rho_{k-1}}{1-\rho_k}, \quad (2.11)$$

$$\bar{\pi}_{k1} = \frac{b_k}{1-\rho_k}, \quad (2.12)$$

$$h_{k1} = \frac{b_k}{1-\rho_{k-1}}, \quad (2.13)$$

$$\upsilon_{k1} = \frac{\Phi_2 ... \Phi_k}{1-\rho_{k-1}} c_{k1}. \quad (2.14)$$

For example, (2.13) gives us average total value of the time during which a query belonging to priority class k stays at the processor, (2.14) is an average value of the full time lost for switching to the priority class k.

A particular case of heterogeneous algorithm could be obtained in the case one assumes, that switching occurs instantly. In this case

$c_{i1} = 0, \ \phi_i = 1.$ and from formulae (2.11) to (2.14) it follows, that

$$\sigma_k \pi_{k1} = \frac{\rho_k}{1-\rho_k}, \quad h_{k1} = \frac{b_k}{1-\rho_{k-1}},$$

which coincides with results obtained by B.V.Gnedenco et al..[4]

# Virtual and Stationary Queues.

The following section is going to reveal common distribution of queuing lengths of takes of each class. The distributions are adequate for any possible time period $t \in (0, \infty)$. Thus, all the below-listed distributions, depending on t, are called virtual or non-stationary distributions. From virtual distribution formulas result the stationary distribution, that is, distribution of queuing lengths, corresponding to the stationary regime.

## Virtual queue

Let $P_m(t)$ be the probability that in time t, $t \in (0, \infty)$ there are $m = (m_1, ..., m_r)$ tasks in the queue, where $m_i$ - the numbers of tasks of class i, i=1,..,r.Let then P(z,t) be the function of these probabilities

$P(z,t) = \sum P_m(t) z^m$, where

$$z^m = z_1^{m_1} \cdots z_r^{m_n}, \quad z = (z_1, ... z_r), \quad 0 \le z_i \le 1,$$

and $p(z,s)$ - is its Laplace transform on t,

$$p(z,s) = \int_0^\infty e^{-st} P(z,t) dt.$$

Let's consider also $[]_k = a_k(1-z_k) + \cdots + a_r(1-z_r).$

Then the common distribution of the queue lengths can be obtained from the following ratio

$$p(z,s) = \frac{1+\sigma\pi(z,s)}{s+\sigma-\sigma\pi(s)}, \quad \sigma\pi(z,s) = \sigma_r \pi_r(z,s)$$

is determined from recurrent equalities

$$\sigma_k \pi_k(z,s) = \sigma_{k-1}\pi_{k-1}(z,s) + \gamma_{k-1}(s,z)\nu_k(z,s) + \frac{h_k(z,s)}{z_k - h_k(s+[]_k)}$$

$$\{\gamma_{k-1}(s,z)\nu_k(s+[]_k)\sigma_{k-1}\pi_{k-1}(s+a_k) - \sigma_k \pi_k(s)\},$$

where

$$h_k(z,s) = \{z_k[1-\beta_k(s+[]_k+\sigma_{k-1})][1+\sigma_{k-1}[\pi_{k-1}(z,s)+\pi_{k-1}(s+[]_k)\nu_k(z,s)]]\}$$

$$\{\gamma_{k-1}(s,z)\nu_k(s+[]_k)\sigma_{k-1}\pi_{k-1}(s+a_k) - \sigma_k \pi_k(s)\},$$

$$\{s+[]_k+\sigma_{k-1}-\sigma_{k-1}[1-\beta_k(s+[]_k+\sigma_{k-1})]\pi_{k-1}(s+[]_k)\nu_k(s+[]_k)\}^{-1},$$

$$\nu_k(z,s) = \frac{1-c_k(s+[]_k)+\sigma_{k-1}[1-\pi_{k-1}(s+[]_k)]}{s+[]_k+\sigma_{k-1}[1-\pi_{k-1}(s+[]_k)]}[1+\sigma_{k-1}\pi_{k-1}(z,s)].$$

The functions are determined in section 2.

These formulas are proved by the "catastrophe" method [4,5].

## Stationary queue.

Let the conditions (2.7) be fulfilled and let denote by $\wp(z)$ the function of stationary distributions $P_m$. It can be shown that there is a limit $P(z,t)$ and

$$\lim_{t\to\infty} P(z,t) = \lim_{s\downarrow 0} s \ \ p(z,s) = \wp(z).$$

Thus, from the formulas of section 3.1 we obtain

$$\wp(z) = \frac{1+\sigma\hat{\pi}(z)}{1+\sigma\pi_1},$$

where $\sigma\hat{\pi}(z) = \sigma_r \pi_r(z, 0)$, and $\pi_1 = \pi_{r1}$ is determined from (2.11) where $k = r$.

As a conclusion it should be noted that, the computing problems connected with the realization of these formulas are more difficult than the realization of homogeneous analogy. At present, we elaborated some effective algorithms of solving the system of equations (2.8)-(2.10), also using the fast-algorithms of Laplace-Stiltjes calculation and it inversion methods.

**Bibliography**

1. Basharin G.P. et al. Queue analysis in computer networks (Russian). Moscow, Nauka, 1989.
2. D.J.Evans and W.U.K.Butt. Dynamic load balancing using task-transfer probabilities. Parallel Computing, 19, p.897-916, 1993.
3. Bogunovic N. Processes scheduling procedure for a class of a real-time computer systems. IEEE Trans. Ind. Electron., v.34, No 1, p.29-34, 1987.
4. Gnedenko B.V. et al. Priority queueing systems (Russian). Moscow, MSU, 1973. Klimov G.P.and Miskoy G.K. Priority queueing systems with orientation (Russian). Moscow, MSU, 1979.