Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

Andreas Brandt[1], Manfred Brandt

# On the stability of the multi-queue multi-server processor sharing with limited service

[1] Institut für Operations Research, Humboldt-Universität zu Berlin, Germany

ZIB-Report 06-25 (May 2006)

# On the stability of the multi-queue multi-server processor sharing with limited service[1]

Andreas Brandt
*Institut für Operations Research, Humboldt-Universität zu Berlin,*
*Spandauer Str. 1, D-10178 Berlin, Germany*
*e-mail: brandt@wiwi.hu-berlin.de*

Manfred Brandt
*Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB),*
*Takustr. 7, D-14195 Berlin, Germany*
*e-mail: brandt@zib.de*

### Abstract

We consider a multi-queue multi-server system with $n$ servers (processors) and $m$ queues. At the system there arrives a stationary and ergodic stream of $m$ different types of requests with service requirements which are served according to the following $k$-limited head of the line processor sharing discipline: The first $k$ requests at the head of the $m$ queues are served in processor sharing by the $n$ processors, where each request may receive at most the capacity of one processor. By means of sample path analysis and Loynes' monotonicity method, a stationary and ergodic state process is constructed, and a necessary as well as a sufficient condition for the stability of the $m$ separate queues are given, which are tight within the class of all stationary ergodic inputs. These conditions lead to tight necessary and sufficient conditions for the whole system, also in case of permanent customers, generalizing an earlier result by the authors for the case of $n = k = 1$.

**Mathematics Subject Classification (MSC 2000):** 60K25, 68M20, 60G10, 60G17, 60G55.

**Keywords:** head of the line processor sharing; thread pools; many queues; many servers; general input; batch arrivals; permanent customers; marked point process; stability condition; Loynes' construction; stationarity; ergodicity.

[1] This work was supported by a grant from the Siemens AG.

# 1 Introduction

In telecommunication systems different processes consisting of requests have to be served. For improving the performance, in modern systems several processors are used in parallel for processing the requests. The system resources, in particular the processor capacities, have to be shared between different types of requests in such a way that on the one hand certain performance characteristics for the different processes can be guaranteed and on the other hand a high system utilization is ensured. Furthermore, the resources should be allocated in a fair manner to the different processes. Various scheduling disciplines are known and implemented for meeting these requirements. In the Round Robin (RR) – also called time sharing – disciplines, cf. e.g. [8], a scheduler allocates a fixed quantum of service, i.e. of processing time, to the requests present under service in a RR manner. RR disciplines ensure that requests with small service requirements have smaller sojourn times compared to those with larger service requirements. For small service quanta the RR disciplines are well approximated by corresponding Processor Sharing (PS) disciplines, which are more convenient for the analysis of time sharing systems, cf. e.g. [6]. PS systems have been studied by many researchers, cf. e.g. [1], [2]-[4], [7], [11], [14], [18]-[24] and the references therein.

The fact that some requests may have to be served sequentially implies for the system architecture that requests have to be queued appropriately and that only the first request in each queue is processed under the PS discipline. This discipline is known as Head of the Line Processor Sharing (HOL-PS) discipline. In case of thread pools also more than one request of a queued thread can be processed simultaneously by the processors. This leads to a generalized HOL-PS discipline considered in this paper, which we call $k$-limited HOL-PS discipline: the first $k$ requests of all queues are served by the processors in the PS mode.

More precisely, in this paper we consider a system consisting of $m \geq 1$ queues and $n \geq 1$ servers (processors), cf. Figure 1.1. At the system there arrives a stream of $m$ types of requests with service requirements. The input is described by a marked point process $\Psi = \{[T_\ell, I_\ell, S_\ell]\}_{\ell=-\infty}^{\infty}$ on the real line with the mark space $\mathbb{K} = \{1, \ldots, m\} \times \mathbb{R}_+$ and $\ldots \leq T_0 \leq 0 < T_1 \leq \ldots$, where $T_\ell$ are the arrival instants of the requests, $I_\ell \in \{1, \ldots, m\}$ indicates the type, i.e. the queue where the request goes to, and $S_\ell \in \mathbb{R}_+$ denotes the required service time of the $\ell$-th request. We assume in the following that $\Psi$ is a stationary and ergodic marked point process, cf. e.g. [5]. Note that batch arrivals are included and that there are no independence assumptions.

The input at queue $i$, i.e. the stream of type-$i$ requests including their

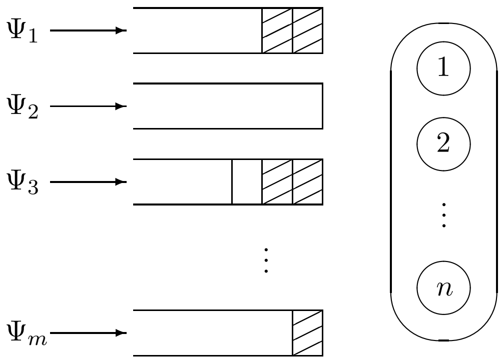

Figure 1.1: *The multi-queue multi-server $k$-limited head of the line processor sharing system with $m$ queues and $n$ processors in case of $k = 2$. The boxes □ correspond to requests, the hatched boxes to requests in service.*

required service times, is given by the stationary ergodic marked point process

$$\Psi_i = \{[T_{i,\ell}, S_{i,\ell}]\}_{\ell=-\infty}^{\infty}\,, \tag{1.1}$$

where $\ldots \leq T_{i,0} \leq 0 < T_{i,1} \leq \ldots$ are the arrival instants of type-$i$ requests and $S_{i,\ell}$ their required service times. For the offered load $\varrho_i$ of the type-$i$ requests it holds

$$\varrho_i = E \sum_{\ell=-\infty}^{\infty} S_{i,\ell} \mathbb{I}\{0 < T_{i,\ell} \leq 1\} = E \int\limits_{(0,1]\times\mathbb{R}_+} y\,\Psi_i(\mathrm{d}(x,y)) = \lambda_i m_{B_i^0},$$

where $\lambda_i = E\,\Psi_i((0,1] \times \mathbb{R}_+)$ is the intensity of $\Psi_i$ and $m_{B_i^0} = ES_i^0$ is the expectation of the service time $S_i^0$ of a typical type-$i$ request, which is given by the Palm distribution of $P(\Psi_i \in (\cdot))$, cf. e.g. [10] formula (1.2.8).

The requests in the queues are served by the processors according to the above mentioned $k$-limited HOL-PS discipline: The first $k \geq 1$ requests of the $m$ queues are served in PS by the $n$ processors. This means, if there are $k_i$ requests in queue $i$, $i = 1, \ldots, m$, then there are altogether

$$b := \sum_{i=1}^{m} \min(k_i, k) \tag{1.2}$$

requests in service, and, if $b$ is positive, each of the first $\min(k_i, k)$ requests from queue $i$ receives the fraction

$$\varphi(b) := \min\left(\frac{n}{b}, 1\right) \tag{1.3}$$

of the capacity of one processor.

Note that the case of $k = 1$ corresponds to the ordinary multi-queue multi-server HOL-PS system, which we simply call multi-queue multi-server HOL-PS system. In case of $k = \infty$ the system is just a $G/G/n - PS$ system, where $G/G$ corresponds to the cumulative arrivals and service times of all requests. If $mk \leq n$ then the $m$ queues act as $G_i/G_i/k/\infty - FCFS$ queues, where $G_i/G_i$ stands for the arrival process and service times of the type-$i$ requests.

The multi-queue single-server HOL-PS ($k = 1$) system with Poisson arrival processes and exponential service times has been considered and analyzed by several authors: In [13] the generating function of the occupancy distribution is derived in case of a completely symmetric system with two queues. In [12] a representation of the joint distribution of the queue length is derived by using power series expansions with respect to the offered load; the established radius of convergence decreases rapidly in the number of queues. For heavy traffic approximations we refer to [9], [13], [16]. HOL-PS systems with limited capacities are analyzed in [9], [17]. In [14] approximations of the mean sojourn time for a PS system with Background Jobs are derived, which covers the HOL-PS model with exponential service times. The multi-queue single-server HOL-PS system with permanent customers is investigated in [2], where partially general service time distributions are considered. The single-queue multi-server system with $k$-limited HOL-PS discipline is a special case of the multi-programmed system given in [21]. For the multi-queue multi-server system with a $k$-limited HOL-PS discipline general results seem not to be known, even for $k = 1$, for our best knowledge.

The paper is organized as follows. In Section 2 by means of Loynes' monotonicity method, for the multi-queue multi-server $k$-limited HOL-PS system with a general stationary and ergodic input, a stationary state process is constructed. In Section 3 a necessary as well as a sufficient condition for the stability of the separate queues (Theorem 3.1, Corollary 3.1) are given. These stability conditions are tight, i.e., they cannot be improved within the class of all stationary and ergodic inputs (Example 3.1). They lead to tight necessary and sufficient conditions for the stability of the whole system (Corollary 3.2), also in case of permanent customers (Corollary 3.3), generalizing a corresponding result for the multi-queue single-server HOL-PS ($k = 1$) system with permanent customers given in [3]. The gap between the necessary and sufficient stability conditions for the system is illustrated (Example 3.2).

## 2 Construction of a stationary state process

Let $R_{i,\ell}(t) \in \mathbb{R}_+$ the residual service time at time $t$ of the $\ell$-th request arrived at queue $i$ before $t$, $\ell = 1, 2, \ldots$, ordered in the reversed order of the numbering of the arrival instants. Thus $R_{i,\ell}(t)$ denotes the residual service time at time $t$ of a $\ell$-th last arrived request at queue $i$ before $t$ where *all* requests arrived before $t$ are counted. Further let

| | | |
|---|---|---|
| $R_i(t) = (R_{i,1}(t), R_{i,2}(t), \ldots)$ | – | infinite vector of the residual service times in queue $i$ at time $t$, |
| $R(t) := (R_1(t), \ldots, R_m(t))$ | – | vector of the residual service times at time $t$. |

We want to construct a stationary state process based on Loynes' monotonicity method, cf. e.g. [5], [15]. Let

$$M_K = \{\psi = \{[t_\ell, i_\ell, s_\ell]\}_{\ell=-\infty}^{\infty} : \ \ldots \leq t_0 \leq 0 < t_1 \leq \ldots, \\ \lim_{\ell \to \pm\infty} t_\ell = \pm\infty, \quad i_\ell \in \{1, \ldots, m\}, \quad s_\ell \in \mathbb{R}_+\} \tag{2.1}$$

be the set of point process realizations where $\Psi$ is concentrated on, i.e. $P(M_K) = 1$. For $\tau > 0$, let

$$r_i^{(\tau)}(t, \psi) = (r_{i,1}^{(\tau)}(t, \psi), r_{i,2}^{(\tau)}(t, \psi), \ldots), \quad i = 1, \ldots, m, \tag{2.2}$$

$$r^{(\tau)}(t, \psi) = (r_1^{(\tau)}(t, \psi), \ldots, r_m^{(\tau)}(t, \psi)) \tag{2.3}$$

be the state of the system at $t$ if it was started at time $t - \tau$ from the empty system with input realization $\psi \in M_K$, where the residual service times of the arrivals until $t - \tau$ are 0 by definition. The workload in queue $i$ at $t$ is given by

$$v_i^{(\tau)}(t, \psi) = \sum_{\ell=1}^{\infty} r_{i,\ell}^{(\tau)}(t, \psi), \quad i = 1, \ldots, m, \tag{2.4}$$

and the number of requests in queue $i$ at $t$ is given by

$$k_i^{(\tau)}(t, \psi) = \sum_{\ell=1}^{\infty} \mathbb{I}\{r_{i,\ell}^{(\tau)}(t, \psi) > 0\}, \quad i = 1, \ldots, m. \tag{2.5}$$

Note that requests arriving or departing at $t$ are not counted in (2.5). As $\varphi(b)$ is non-increasing with respect to $b$, cf. (1.3), it follows from the dynamics of the $k$-limited HOL-PS discipline, in particular from the fact that all requests under service receive an equal fraction of the processor capacity, that the vector of the residual service times $r^{(\tau)}(t,\psi)$, i.e. each partial component $r^{(\tau)}_{i,\ell}(t,\psi)$, is non-decreasing with respect to $\tau$, which is crucial in Loynes' method. Because of this monotonicity, the limit as $\tau \to \infty$ exists:

$$r_{i,\ell}(t,\psi) := \lim_{\tau\to\infty} r^{(\tau)}_{i,\ell}(t,\psi), \quad i = 1,\ldots,m, \quad \ell = 1,2,\ldots, \tag{2.6}$$

$$r(t,\psi) := \lim_{\tau\to\infty} r^{(\tau)}(t,\psi). \tag{2.7}$$

The state $r(t,\psi)$ corresponds to the system state at $t$ if the system was started at time $-\infty$ from the initial state where all residual service times are 0. Of course, the number of requests in queue $i$

$$k_i(t,\psi) := \sum_{\ell=1}^{\infty} \mathbb{I}\{r_{i,\ell}(t,\psi) > 0\}, \quad i = 1,\ldots,m, \tag{2.8}$$

may be infinite for some queues. However, $r(t,\psi)$ satisfies the system dynamics due to continuity arguments. From now on let

$$R(t) := r(t,\Psi), \quad t \in \mathbb{R}, \tag{2.9}$$

which is a stationary and ergodic process due to the monotonicity property and since the residual service times of the arrivals until $t-\tau$ are 0 by definition. Note that $R(t)$, $t \in \mathbb{R}$, is the minimal stationary state process. Let

$$K_i(t) := k_i(t,\Psi), \quad t \in \mathbb{R}, \tag{2.10}$$

the corresponding stationary number of requests in queue $i$ at $t$.

# 3 Stability conditions

A reasonable definition of stability for the model considered, cf. Theorem 3.1 (i), is the following one:

**Definition 3.1** *Queue $i \in \{1,\ldots,m\}$ is stable if $P(K_i(0) < k) > 0$. The processor sharing system is stable if $P(K_i(0) < k) > 0$ for $i = 1,\ldots,m$.*

Let

$$p_i := P(K_i(0) \geq k), \quad i = 1, \ldots, m, \tag{3.1}$$

be the probability that at time 0 there are at least $k$ type-$i$ requests in the system. Note that in case of $k = 1$, i.e. in case of the ordinary HOL-PS system, $p_i$ is the probability that queue $i$ is non-empty at time 0. Because of the stationarity and ergodicity of $R(t)$, it holds

$$p_i = \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathbb{I}\{K_i(x) \geq k\} \mathrm{d}x \quad \text{P-a.s.}, \quad i = 1, \ldots, m. \tag{3.2}$$

By Definition 3.1 queue $i$ is stable if and only if $p_i < 1$.

**Theorem 3.1** *Let*

$$\varrho_1 \geq \varrho_2 \geq \ldots \geq \varrho_m > 0. \tag{3.3}$$

*For the minimal stationary state process $R(t)$ given by* (2.9), (2.7) *it holds:*

*(i) If queue $i$ is stable then queue $j$ is stable if $\varrho_j \leq \varrho_i$.*
*If queue $i$ is unstable then queue $j$ is unstable if $\varrho_j \geq \varrho_i$.*

*(ii) Queue $i \in \{1, \ldots, m\}$ is stable if there exists a $h \in \{ik, ik+1, \ldots, mk\}$ such that*

$$(h/k)\varrho_i + (\lceil h/k \rceil - h/k)\varrho_{\lceil h/k \rceil} + \sum_{j=\lceil h/k \rceil+1}^{m} \varrho_j < \min(n, h). \tag{3.4}$$

*(iii) If queue $i \in \{1, \ldots, m\}$ is stable then*

$$\varrho_i < k, \qquad i\varrho_i + \sum_{j=i+1}^{m} \varrho_j \leq n \tag{3.5}$$

*with equality only in case of $\varrho_i = \varrho_1$.*

**Proof** If the system is non-empty at $t$ then any served request receives the random fraction

$$C^*(t) = \min\left( \frac{n}{\sum_{j=1}^{m} \min(K_j(t), k)}, 1 \right) \tag{3.6}$$

of the capacity of one processor at $t$, cf. (1.2), (1.3). For technical convenience, let $C^*(t) := 1$ if the system is empty at $t$. Thus queue $i$ receives the random multiple

$$C_i(t) = \min(K_i(t), k)C^*(t), \quad i = 1, \ldots, m, \tag{3.7}$$

of the capacity of one processor at time $t$. Because of the stationarity and ergodicity of $R(t)$, for the mean multiple of the processor capacity received by queue $i$ it holds

$$EC_i(0) = \lim_{t\to\infty} \frac{1}{t} \int_0^t C_i(x)\mathrm{d}x \quad \text{P-a.s.}, \quad i = 1, \ldots, m. \tag{3.8}$$

Let

$$\varrho^* := kEC^*(0). \tag{3.9}$$

Then from (3.1), (3.6), (3.7), (3.9) it follows

$$\varrho^* - k(1-p_i) \le EC_i(0) \le \varrho^* - \min\Big(\frac{n}{mk}, 1\Big)(1-p_i), \quad i = 1, \ldots, m. \tag{3.10}$$

We use Loynes' construction (2.1)–(2.7). Let $\tau > 0$ be fixed. Starting the dynamics of the processor sharing system at time $-\tau$ from the empty system then

$$c_i^{(\tau)}(\Psi) := v_i^{(\tau)}(0{+}0, \Psi) - v_i^{(\tau+1)}(1{+}0, \Psi) + \sum_{\ell=-\infty}^{\infty} S_{i,\ell}\mathbb{I}\{0{<}T_{i,\ell}{\le}1\}, \quad i = 1, \ldots, m, \tag{3.11}$$

is just the amount of service that receive the type-$i$ requests during the interval $(0, 1]$ by the processors. Taking expectations we find

$$Ec_i^{(\tau)}(\Psi) = Ev_i^{(\tau)}(0{+}0, \Psi) - Ev_i^{(\tau+1)}(1{+}0, \Psi) + \varrho_i. \tag{3.12}$$

By the stationarity of $\Psi$ and since $v_i^{(\tau)}(0 + 0, \Psi)$ is non-decreasing with respect to $\tau$, it holds

$$Ev_i^{(\tau+1)}(1{+}0, \Psi) = Ev_i^{(\tau+1)}(0{+}0, \Psi) \ge Ev_i^{(\tau)}(0{+}0, \Psi).$$

Thus (3.12) yields

$$Ec_i^{(\tau)}(\Psi) \le \varrho_i, \quad i = 1, \ldots, m.$$

By the stationarity of $\Psi$, taking the limit as $\tau \to \infty$ we obtain

$$EC_i(0) \le \varrho_i, \quad i = 1, \dots, m. \tag{3.13}$$

Let $p_i < 1$. Because of (3.2), then we conclude that in any neighborhood of infinity queue $i$ possesses periods where less than $k$ requests are in queue $i$. Thus, in view of the stationarity of $\Psi$ and of the system dynamics, all requests arriving at queue $i$ will be served. Since $\int_0^t C_i(x)\mathrm{d}x$ is the amount of service received by queue $i$ during the interval $(0, t]$, we find

$$\varrho_i = \lim_{t\to\infty} \frac{1}{t} \int\limits_{(0,t]\times\mathbb{R}_+} y\Psi_i(\mathrm{d}(x,y)) = \lim_{t\to\infty} \frac{1}{t} \int\limits_0^t C_i(x)\,\mathrm{d}x = EC_i(0), \tag{3.14}$$

i.e., $\varrho_i$ is just the mean multiple of the processor capacity received by queue $i$ provided $p_i < 1$.

If $\varrho_i < \varrho^*$ then from (3.13) it follows $EC_i(0) < \varrho^*$, and hence from (3.10) we find $p_i < 1$, i.e., queue $i$ is stable. Let $\varrho_i \ge \varrho^*$. Assuming $p_i < 1$, from (3.14) it follows $EC_i(0) = \varrho_i \ge \varrho^*$, and hence (3.10) provides a contradiction. Therefore $p_i = 1$, and queue $i$ is unstable if $\varrho_i \ge \varrho^*$. Summarizing, it holds the following necessary and sufficient stability condition for queue $i$: queue $i$ is stable if and only if

$$\varrho_i < \varrho^*. \tag{3.15}$$

This stability condition implies (i).

From (3.15), (3.14), (3.10), (3.7) and (3.6) we find

$$\begin{aligned}\sum_{j=1}^m \min(\varrho_j, \varrho^*) &= \sum_{j=1}^m (\mathbb{I}\{p_j < 1\}\varrho_j + \mathbb{I}\{p_j = 1\}\varrho^*) = \sum_{j=1}^m EC_j(0) \\ &= E\Big[\min\Big(n, \sum_{j=1}^m \min(K_j(0), k)\Big)\Big].\end{aligned} \tag{3.16}$$

Let us assume $p_i = 1$. Because of (3.3) and (i), then queue $j$ is unstable for $j \in \{1, \dots, i\}$, i.e.

$$1 = p_j = P(K_j(0) \ge k), \quad j = 1, \dots, i. \tag{3.17}$$

Hence from (3.16) it follows

$$\sum_{j=1}^m \min(\varrho_j, \varrho^*) \ge E\Big[\min\Big(n, \sum_{j=1}^i \min(K_j(0), k)\Big)\Big] = \min(n, ik). \tag{3.18}$$

Taking into account (3.3) and (3.15), thus we find

$$i\varrho_i + \sum_{j=i+1}^{m} \varrho_j = \sum_{j=1}^{m} \min(\varrho_j, \varrho_i) \geq \sum_{j=1}^{m} \min(\varrho_j, \varrho^*) \geq \min(n, ik). \quad (3.19)$$

This estimate can be tightened as follows.

Firstly, we consider the modified system with $mk$ queues and $nk$ processors where the input at queue $j$ is given by

$$\Psi_j^{(1)} := \Psi_{\lceil j/k \rceil}, \quad j = 1, \ldots, mk. \quad (3.20)$$

Thus for the offered load at queue $j$ in the modified system it holds

$$\varrho_j^{(1)} = \varrho_{\lceil j/k \rceil}, \quad j = 1, \ldots, mk. \quad (3.21)$$

Note that the input at the modified system consists of $k$ copies of the input at the original system and that moreover the dynamics of the modified system consist of $k$ copies of the dynamics of the original system. Hence queue $j$ in the modified system is stable if and only if queue $\lceil j/k \rceil$ in the original system is stable, in particular, in the modified system queue $ik$ is unstable.

Let $h \in \{ik, ik+1, \ldots, mk\}$. Secondly, we scale the service times per queue in the modified system such that for the offered load $\varrho_j^{(2)}$ of type-$j$ requests in the newly modified system it holds

$$\varrho_j^{(2)} := \begin{cases} \varrho_j^{(1)}, & j = 1, \ldots, ik, \\ \varrho_{ik}^{(1)}, & j = ik+1, \ldots, h, \\ \varrho_j^{(1)}, & j = h+1, \ldots, mk. \end{cases} \quad (3.22)$$

In view of $\varrho_j^{(2)} \geq \varrho_j^{(1)}$ for $j = 1, \ldots, mk$, from the monotonicity property of Loynes' construction with respect to the service times it follows that also in the newly modified system queue $ik$ is unstable. Because of (3.22), (3.21), (3.3) and (i), thus queue $j$ is unstable for $j \in \{1, \ldots, h\}$ in the newly modified system. Hence (3.19) applied to the newly modified system provides

$$h\varrho_h^{(2)} + \sum_{j=h+1}^{mk} \varrho_j^{(2)} \geq \min(nk, hk).$$

In view of (3.22), (3.21), thus we obtain

$$h\varrho_i + \sum_{j=h+1}^{mk} \varrho_{\lceil j/k \rceil} \geq \min(nk, hk),$$

which is equivalent to the following tightening of (3.19):

$$(h/k)\varrho_i + (\lceil h/k\rceil - h/k)\varrho_{\lceil h/k\rceil} + \sum_{j=\lceil h/k\rceil+1}^{m} \varrho_j \geq \min(n,h). \tag{3.23}$$

Thus queue $i$ in the original system is stable if there exists a $h \in \{ik,\ldots,mk\}$ such that

$$(h/k)\varrho_i + (\lceil h/k\rceil - h/k)\varrho_{\lceil h/k\rceil} + \sum_{j=\lceil h/k\rceil+1}^{m} \varrho_j < \min(n,h), \tag{3.24}$$

which is part (ii).

Let $p_i < 1$. From (3.15), (3.9) and (3.6) it follows the first part of (iii):

$$\varrho_i < \varrho^* \leq k. \tag{3.25}$$

Because of (3.16), it holds

$$\sum_{j=1}^{m} \min(\varrho_j, \varrho^*) \leq n, \tag{3.26}$$

and in view of (3.3) and (3.15), we obtain the second part of (iii):

$$i\varrho_i + \sum_{j=i+1}^{m} \varrho_j = \sum_{j=1}^{m} \min(\varrho_j, \varrho_i) \leq \sum_{j=1}^{m} \min(\varrho_j, \varrho^*) \leq n \tag{3.27}$$

with equality of the last both sums only if $\varrho_i = \varrho_1$ as the first summands are equal only if $\varrho_i = \varrho_1$.

□

Choosing $h = ik$ for $n < ik$, $h = n$ for $ik \leq n < mk$ and $h = mk$ for $mk \leq n$ in Theorem 3.1, respectively, we obtain the following corollary:

**Corollary 3.1** *Let* $\varrho_1 \geq \varrho_2 \geq \ldots \geq \varrho_m > 0$.

*(i) Let* $n < ik$. *Queue* $i$ *is stable if*

$$i\varrho_i + \sum_{j=i+1}^{m} \varrho_j < n, \tag{3.28}$$

*and if queue* $i$ *is stable then*

$$i\varrho_i + \sum_{j=i+1}^{m} \varrho_j \leq n \tag{3.29}$$

*with equality only in case of* $\varrho_i = \varrho_1$.

(ii) *Let $ik \leq n < mk$. Queue $i$ is stable if*

$$(n/k)\varrho_i + (\lceil n/k \rceil - n/k)\varrho_{\lceil n/k \rceil} + \sum_{j=\lceil n/k \rceil+1}^{m} \varrho_j < n, \tag{3.30}$$

*and if queue $i$ is stable then*

$$\varrho_i < k, \qquad i\varrho_i + \sum_{j=i+1}^{m} \varrho_j \leq n \tag{3.31}$$

*with equality only in case of $\varrho_i = \varrho_1$.*

(iii) *Let $mk \leq n$. Queue $i$ is stable if and only if*

$$\varrho_i < k. \tag{3.32}$$

The following example tells us that the sufficient as well as the necessary stability conditions given in Corollary 3.1 are tight, i.e., they cannot be improved within the class of all stationary and ergodic inputs.

**Example 3.1** *Let $n < mk$ and $\varrho_1 \geq \varrho_2 \geq \ldots \geq \varrho_m > 0$ be given and $U$ be uniformly distributed on $[0, 1]$.*

(i) *Consider the case of batch arrivals of size $mk$ where at the time instants $\ell + U$, $\ell \in \mathbb{Z}$, $k$ requests with service time $\varrho_j/k$ arrive at queue $j$, $j \in \{1, \ldots, m\}$.*

*In this case the service of an arriving batch at queue $i$ takes at least the time $\sum_{h=i}^{m}(\varrho_h - \varrho_{h+1}) \max(h/n, 1/k)$ where $\varrho_{m+1} := 0$. The stability of queue $i$ implies that this duration has to be less than 1, i.e.* (3.28) *if $n < ik$ and* (3.30) *if $ik \leq n < mk$. Thus the sufficient stability conditions are tight.*

(ii) *Consider the case where at the time instants $\ell + \sum_{h=1}^{j-1} \varrho_h + U$, $\ell \in \mathbb{Z}$, $j \in \{1, \ldots, m\}$, a request with service time $\varrho_j$ arrives at queue $j$.*

*Let $n < ik$, $\varrho_i = \varrho_1$ and* (3.29) *be fulfilled or let $ik \leq n < mk$, $\varrho_i = \varrho_1$ and* (3.31) *be fulfilled. Then it holds $\varrho_1 = \varrho_i < k$ and $\sum_{h=1}^{m} \varrho_h \leq n$. Therefore at any time $t$ there are altogether at most $\lceil \sum_{h=1}^{m} \varrho_h \rceil \leq n$ requests which arrived at any queue $j$ during $[t - \varrho_j, t)$ where at most $k$ requests arrived at the same queue due to $\varrho_j < k$, and thus all these requests receive the capacity of one processor. Hence queue $i$ is stable* *as $\varrho_i < k$. Thus the necessary stability condition* (3.29) *is tight, and the necessary stability conditions* (3.31) *are tight in case of $\varrho_i = \varrho_1$.*

*Let $ik \le n < mk$, $\varrho_i < \varrho_1$ and let* (3.31) *be fulfilled. Assume that queue $i$ is unstable. In view of Theorem 3.1 (i), then the first $i$ queues are unstable. In case of $\sum_{h=i+1}^{m} \varrho_h \le n - ik$ at any time $t$ there are altogether at most $\lceil \sum_{h=i+1}^{m} \varrho_h \rceil \le n - ik$ requests which arrived at any queue $j$ during $[t - \varrho_j, t)$ for any $j > i$ where at most $k$ requests arrived at the same queue due to $\varrho_j < k$ for $j > i$, and thus all served requests in the system receive the capacity of one processor. Thus queue $i$ receives almost surely the capacity of $k$ processors in contradiction to $\varrho_i < k$ and* (3.13)*. In case of $\sum_{h=i+1}^{m} \varrho_h > n - ik$ at any time $t$ there are altogether at least $\lfloor \sum_{h=i+1}^{m} \varrho_h \rfloor \ge n - ik$ requests which arrived at any queue $j$ during $[t - \varrho_j, t)$ for any $j > i$ where at most $k$ requests arrived at the same queue due to $\varrho_j < k$ for $j > i$. As any served request receives at most the capacity of one processor thus there are almost surely at least $n$ served requests in the system, and thus the $n$ processors are almost surely busy. Hence each of the queues $j \le i$ receives at least the mean multiple $(n - \sum_{h=i+1}^{m} \varrho_h)/i > \varrho_i$ of the capacity of one processor in contradiction to* (3.13) *applied to queue $i$. Thus queue $i$ is stable, and the necessary stability conditions* (3.31) *are tight in case of $\varrho_i < \varrho_1$, too.*

Choosing $i := 1$ in Corollary 3.1 provides tight stability conditions for the whole system:

**Corollary 3.2** *Let $\varrho_1 \ge \varrho_2 \ge \ldots \ge \varrho_m > 0$.*

*(i) Let $n < mk$. The system is stable if*

$$(n/k)\varrho_1 + (\lceil n/k \rceil - n/k)\varrho_{\lceil n/k \rceil} + \sum_{j=\lceil n/k \rceil + 1}^{m} \varrho_j < n, \tag{3.33}$$

*and if the system is stable then*

$$\varrho_1 < k, \qquad \sum_{j=1}^{m} \varrho_j \le n. \tag{3.34}$$

*(ii) Let $mk \le n$. The system is stable if and only if*

$$\varrho_1 < k. \tag{3.35}$$

**Example 3.2** *Let $m = 3$, $n = 2$, $k = 1$ and $\varrho_1 \geq \varrho_2 = 0.7$, $\varrho_3 = 0.5$. Then the sufficient stability condition* (3.33) *reads $\varrho_1 < 0.75$, and the necessary stability conditions* (3.34) *reads $\varrho_1 \leq 0.8$. Therefore the system is stable if $\varrho_1 < 0.75$ and unstable if $\varrho_1 > 0.8$.*

*Simulations of the HOL-PS system with Poisson arrival streams of intensity $\varrho_i$, $i = 1, 2, 3$, and exponential service times with mean* 1 *provide the approximate necessary and sufficient stability condition $\varrho_1 < 0.76$ in the special case of the Markov model with equal mean service times.*

Consider the modified $k$-limited HOL-PS system where in each of the first $l < m$ queues there are $k$ permanent customers, i.e., where in each of the first $l$ queues there are $k$ requests with infinite service requirement. In this case a reasonable definition of stability for the whole system would be the stability of the remaining $m - l$ queues, cf. [3].

We model the case of permanent customers as described above by choosing the offered loads for the first $l$ queues sufficiently large. Thus we may assume that $\varrho_1 \geq \ldots \geq \varrho_l > \varrho_{l+1} \geq \ldots \geq \varrho_m > 0$. Choosing now $i := l + 1$ in Corollary 3.1 provides tight stability conditions for the whole multi-queue multi-server $k$-limited HOL-PS system with permanent customers, generalizing a corresponding result for the ordinary multi-queue single-server HOL-PS system with permanent customers given in [3] Theorem 3.5 (iv):

**Corollary 3.3** *Consider the modified processor sharing system where in each of the first $l$ queues there are $k$ permanent customers and $0 < l < m$. Let $\varrho_{l+1} \geq \varrho_{l+2} \geq \ldots \geq \varrho_m > 0$.*

*(i) Let $(l + 1)k < n < mk$. The system is stable if*

$$(n/k)\varrho_{l+1} + (\lceil n/k \rceil - n/k)\varrho_{\lceil n/k \rceil} + \sum_{j=\lceil n/k \rceil+1}^{m} \varrho_j < n, \tag{3.36}$$

*and if the system is stable then*

$$\varrho_{l+1} < k, \qquad l\varrho_{l+1} + \sum_{j=l+1}^{m} \varrho_j < n. \tag{3.37}$$

*(ii) Let $n \leq (l + 1)k$ or $mk \leq n$. The system is stable if and only if* (3.37) *holds.*

## References

[1] Boxma, O.J., Groenendijk, W.P., Waiting times in discrete-time-cyclic-service systems. IEEE Trans. Commun. 36 (1988) 164–170.

[2] Brandt, A., Brandt, M., On the sojourn times for many-queue head-of-the-line processor-sharing systems with permanent customers. Math. Methods Oper. Res. 47 (1998) 181–220.

[3] Brandt, A., Brandt, M., A note on the stability of the many-queue head-of-the-line processor-sharing system with permanent customers. Queueing Syst. 32 (1999) 363–381.

[4] Brandt, A., Brandt, M., A sample path relation for the sojourn times in $G/G/1-PS$ systems and its applications. Queueing Syst. 52 (2006) 281–286.

[5] Brandt, A., Franken, P., Lisek, B., *Stationary Stochastic Models.* Akademie-Verlag, Berlin; Wiley, Chichester 1990.

[6] Coffman, E.G., Muntz, R.R., Trotter, H., Waiting time distributions for processor-sharing systems. J. Assoc. Comput. Mach. 17 (1970) 123–130.

[7] Cohen, J.W., The multiple phase service network with generalized processor sharing. Acta Inf. 12 (1979) 245–284.

[8] Doshi, B.T., Rege, K. M., Analysis of a multistage queue. AT&T Bell Lab. Techn. J. 64 (1985) 369–390.

[9] Fendick, K.W., Rodrigues, M.A., A heavy-traffic comparison of shared and segregated buffer schemes for queues with the head-of-line processor-sharing discipline. Queueing Syst. 9 (1991) 163–190.

[10] Franken, P., König, D., Arndt, U., Schmidt, V., *Queues and Point Processes.* Akademie-Verlag, Berlin; Wiley, Chichester 1982.

[11] Guillemin, F., Robert, P., Zwart, B., Tail asymptotics for processor-sharing queues. Adv. Appl. Probab. 36 (2004) 525–543.

[12] Hooghiemstra, G., Keane, M., van de Ree, S., Power series for stationary distributions of coupled processor models. SIAM J. Appl. Math. 48 (1988) 1159–1166.

[13] Konheim, A.G., Meilijson, I., Melkman, A., Processor-sharing of two parallel lines. J. Appl. Probab. 18 (1981) 952–956.

[14] Leung, K.K., Performance analysis of a processor sharing policy with interactive and background jobs. IFIP Transactions Vol. C-5 (1991) 189–207.

[15] Loynes, R.M., The stability of a queue with nonindependent interarrival and service times. Proc. Camb. Philos. Soc. 58 (1962) 497–520.

[16] Morrison, J.A., Diffusion approximation for head-of-the-line processor sharing for two parallel queues. SIAM J. App. Math. 53 (1993) 471–490.

[17] Morrison, J.A., Head of the line processor sharing for many symmetric queues with finite capacity. Queueing Syst. 14 (1993) 215–237.

[18] Núñez-Queija, R., Sojourn times in a processor sharing queue with service interruptions. Queueing Syst. 34 (2000) 351–386.

[19] Ott, T.J., The sojourn-time distribution in the $M/G/1$ queue with processor sharing. J. Appl. Probab. 21 (1984) 360–378.

[20] Ramaswami, V., The sojourn time in the $GI/M/1$ queue with processor sharing. J. Appl. Probab. 21 (1984) 437–442.

[21] Rege, K.M., Sengupta, B., Sojourn time distribution in a multiprogrammed computer system. AT&T Bell Lab. Techn. J. 64 (1985) 1077–1090.

[22] Sericola, B., Guillemin, F., Boyer, J., Sojourn times in the $M/PH/1$ processor sharing queue. Queueing Syst. 50 (2005) 109–130.

[23] van Uitert, M., Borst, S.C., A reduced-load equivalence for generalised processor sharing networks with long-tailed input flows. Queueing Syst. 41 (2002) 123–163.

[24] Yashkov, S.F., Mathematical problems in the theory of shared-processor systems. Itogi Nauki Tekh., Ser. Teor. Veroyatn., Mat. Sta., Teor. Kibern. 29 (1990) 3–82.