Real Time Systems with Nonpreemptive Priorities and Shortage of Maintenance Teams

Joseph Kreimer

Department of Industrial Engineering and Management, Ben-Gurion University of the Negev,P.O.Box 653, Beer-Sheva8499000, Israel.

ABSTRACT:*We consider a real time data acquisition and processing multiserver system with identical servers (such as unmanned aerial vehicles, machine controllers, overhearing devices, medical monitoring units, etc.) which can be maintained/programmed for different kinds of activities (e.g. passive or active). This system provides a service for real time tasks arriving via several channels (such as surveillance regions, assembly lines, communication channels, etc.) and involves maintenance.*

We focus on the worst case analysis of the system with shortage of maintenance teams, exponentially distributed time to failure and maintenance times. We consider the model with nonpreemptive priorities and provide balance equations for steady state probabilities and various performance measures, when both operation and maintenance times are exponentially distributed.

KEYWORDS - *performance, non-preeemptive priority, queueing, real time system,unmanned aerial vehicles*

I. Introduction

Real time systems (RTS) are imbedded in most modern technological structures, such as robotic systems, production control systems, radars, self-guided missiles, reconnaissance, aircraft, etc. These systems are widely used for the monitoring and control of physical processes.
The sustained demands of the environments in which RTS operate pose relatively rigid requirements (such as time constraints) on their performance.
In RTS a calculation that uses temporally invalid data or an action performed too early/late, may be sometimes harmful – even if such a calculation or action is functionally correct. These systems are often associated with applications, in which human lives or expensive machinery may be at stake.
There exists a rich literature covering RTS and different scientific communities are treating various problems in this area. We will site only a small portion of it.
Liu and Layland in [1] presented analytic combinatorial analysis for multiprogramscheduling on a single processor. Dhall and Liu [2] treated the problem of specifying an order in which the requests made by a set ofperiodic real-time jobs are to be executed by a multiprocessor computing system, with the goal of meeting all the deadlines with a minimum number of processors. Xio et al. in [3] presented a general cost model and an optimal solution algorithm achieving required performance objective for redundant RTS.
Metaheuristic methods, such as Tabu Search [4] and Greedy Randomized Adaptive Search Procedure [5] were proposed during the last two decades of the previous century. A comprehensive survey on real-time decision problems in OR perspective is given in [6]. Good survey on applications of Artificial Intelligence to RTS is given in [7].
We focus on RTS with a zero deadline for the beginning of job processing. Queueing of jobs in such systems is impossible, since jobs are executed immediately upon arrival, conditional on system availability. That part of the job which is not processed immediately is lost forever and cannot be served later.
The following works treated this type of RTS: Kreimer and Mehrez ([8] and [9]) have shown that the non-mix policy of never relieving an operative server maximizes the availability of a single channel RTS involving preventive maintenance and working in general regime with any arrival pattern under consideration and constant service and maintenance times. [10] and [11], [12] studied muliserver and multichannel (with identical servers and channels) RTS *without priorities* (with ample and limited maintenance facilities respectively), working

under maximum load regime as finite source queues (see [13]). Kreimer [14] applied the two dimensional birth and death processes in worst case analysis of a multiserver RTS (with ample and limited maintenance facilities respectively) with two different channels, when both service and maintenance times were exponentially distributed. Ianovsky and Kreimer [15] obtained optimal assignment/routing probabilities to maximize the availability of RTS with limited maintenance facilities, for large number of servers and two different channels. In [16] the steady state probabilities for RTS with arbitrary number of different channels operating under a maximum load regime were obtained. In [17] and [18] multiserver and multichannel RTS working in general regime (with ample and limited maintenance facilities respectively) are considered. In [19] a single-channel RTS with two different types of activities was studied. In [20] we computed analytically (for exponentially distributed service times) and via Cross Entropy [21],[22], [23] simulation approach (for generally distributed service times) optimal routing probabilities for RTS with ample maintenance facilities. In [24], [25] and [105] RTS with preemptive priorities are treated. In [27] RTS with nonpreemptive priorities and ample maintenance facilities was studied.

The work presented here is a generalization of results obtained in [19] and [27] for RTS with several types of activities and limited maintenance facilities. It deals with multiserver RTS, providing the service to the requests of real-time jobs arriving via several channels. Servers are identical, but may be maintained/programmed for several kinds of activities. We provide balance equations describing the model with nonpreemptive priorities operating under a maximum load and show how to compute various performance measures, when both operation and maintenance times are exponentially distributed.

The paper is organized as follows: In Section 2, the description of the model is presented. Section 3 provides balance equations for RTS with nonpreemptive priorities. Section 4 is devoted to description of various performance characteristics. Finally, Section 5 contains conclusions.

II. The Model

The most important characteristics of RTS with a zero deadline for the beginning of job processing are summarized in Kreimer and Mehrez [24] as follows:

(i) Jobs/data acquisition and execution are as fast as the data arrival rate.
(ii) Jobs are executed immediately upon arrival, conditional on system availability. Between jobs arrival and their execution, delays are negligible.
(iii) That part of the job which is not executed in real time is lost forever. Storage of non-completed jobs or their parts is impossible. Therefore, queues of jobs do not exist in such an RTS.
Nevertheless, some important elements of queueing theory can be applied in analysis of these RTS.

We consider a multiserver multichannel RTS consisting of N identical servers (e.g. UAVs, machine controllers, overhearing devices, medical monitoring devices, etc.)that provide service for the requests of real time jobs, arriving via r identical channels (e.g. surveillance regions, assembly lines, communication channels, hospital patients, etc.).

Each server can serve different channels, but not simultaneously. There is exactly one request in each channel at any moment (there are no additional job arrivals to busy channels), and therefore one server at most is used (with others being in maintenance or on stand-by or providing the service to another channel) to execute the job in this channel at any given time. Therefore, the total number of jobs in the system is r (as a number of channels). Different parts of the same job can be served by different servers. Any part of the job that is not executed immediately is lost.

It is assumed that there are K ($K<N$) identical maintenance teams available to repair (with repair times R_i beingindependent identically distributed random variables – i.i.d.r.v.) broken servers. Thus we have a shortage of maintenance teams, if the number of broken servers is more than K, and some of broken servers must wait for maintenance. A fixed server may be of one of m kinds (e.g. of activity or quality) with probabilities p_i, $i=1,\dots,m$ respectively. These probabilities can be used sometimes as control parameters. It is worthwhile to note, that onlyafterthe maintenance is completed, the type of fixed server can be detected. The fixed server of i-th type is operative for a period of time S_i before requiring R_i hours of repair. Both S_i and R_i areindependent exponentially distributed random values with parameters μ_i ($i=1,\dots m$) and λ respectively. The duration R_i of

maintenance does not depend on the server's type (neither before nor after the maintenance).After maintenance is completed, the fixed server will either be on stand-by or providing service to one of channels.

The system works under a maximum load of nonstop data arrival, which is equivalent to the case of a unique job of infinite duration in each one of r channels. This kind of operation is typical in high performance data acquisition and control systems. Decisions based on the *nonstop* data flow arriving via all $r$channels must be made *in real time*. If, during some period of time of length T, there is no available server to serve one of the channels, it means that the part of the job of length T is lost forever.

Our purpose is to provide balance equations for steady-state probabilities of this system, its availability and other performance characteristics.

III. Balance Equations

We suppose that servers of first activity type have the highest priority, servers of the second activity type are after them, and so on, finally, servers of the m-th activity type have the lowest priority. Operating server is not interrupted, independently of its priority. When the operating server must be repaired, the fixed server with highest priority takes its place.

We denote $\{(n_1,k_1),\dots,(n_m,k_m)\}$ the state of the system, where n_i ($i=1,\dots,m$) is the total number of fixed servers of i-th activity type and k_i ($i=1,\dots,m$) is the number of operating (providing service) fixed servers of i-th activity type respectively, and $p\{(n_1,k_1),\dots,(n_m,k_m)\}$ the corresponding steady state probability. It is clear that $n_1+\dots+n_m \le N$, $n_i \ge k_i \ge 0$, $\sum_{i=1}^{m} k_i = \min\left[r, \sum_{i=1}^{m} n_i\right]$.

Theorem. Steady state probabilities for the model with nonpreemptive priorities with shortage of maintenance teams can be found from the following system of linear balance equations:

$$\left[\sum_{i=1}^{m} k_i\mu_i + \min[K,\left(N-\sum_{j=1}^{m} n_j\right)]\lambda\right] p\{(n_1,k_1),(n_2,k_2),\dots,(n_m,k_m)\} = \min\left(r-\sum_{i=1}^{m} k_i, 1\right) *$$

$$* \sum_{j=1}^{m} (k_j+1)\mu_j\, p\{(n_1,k_1),\dots,(n_{j-1},k_{j-1}),(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\dots,(n_m,k_m)\} +$$

$$+\left(1-\min\left(r-\sum_{i=1}^{m} k_i, 1\right)\right)\sum_{j=1}^{m}\sum_{i=1}^{m}(k_j+1-\delta_{ij})\mu_j\left(1-\min\left(\sum_{s=1}^{i-1}(n_s-k_s),1\right)\right) *$$

$$* p\{(n_1,k_1),\dots,(n_{i-1},k_{i-1}),(n_i+\delta_{ij},k_i-1+\delta_{ij}),(n_{i+1},k_{i+1}),\dots,(n_{j-1},k_{j-1}),$$

$$(n_j+1,k_j+1-\delta_{ij}),(n_{j+1},k_{j+1}),\dots,(n_m,k_m)\} +$$

$$+\left(1-\min\left(\sum_{i=1}^{m}(n_i-k_i),1\right)\right)\sum_{j=1}^{m}\min[K,\left(N-\sum_{i=1}^{m} n_i+1\right)]\lambda p_j *$$

$$* p\{(n_1,k_1),\dots,(n_{j-1},k_{j-1}),(n_j-1,k_j-1),(n_{j+1},k_{j+1}),\dots,(n_m,k_m)\} +$$

$$+\min\left(\sum_{i=1}^{m}(n_i-k_i),1\right)\sum_{j=1}^{m}\min[K,\left(N-\sum_{i=1}^{m} n_i+1\right)]\lambda p_j *$$

$$* p\{(n_1,k_1),\dots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\dots,(n_m,k_m)\}, \qquad (1)$$

where $\{(n_1,k_1),\ldots,(n_m,k_m)\}$ is any feasible state of the system, $\delta_{ij}=\begin{cases}0, & if \quad i\neq j\\ 1, & if \quad i=j\end{cases}$.

and

$$\sum_{\{(n_1,k_1),\ldots,(n_m,k_m)\}\in E} p\{(n_1,k_1),\ldots,(n_m,k_m)\}=1, \tag{2}$$

where $E=\left\{\{(n_1,k_1),\ldots,(n_m,k_m)\}: 0\le k_i\le n_i, i=\overline{1,m}, \sum_{i=1}^{m} n_i\le N, \sum_{i=1}^{m} k_i=\min\left(r,\sum_{j=1}^{m} n_j\right)\right\}$

is the set of all feasible states of the system.

<u>*Proof*</u>:

1) First we consider the case $r\ge N$. Then it is clear that $n_i=k_i$, $i=\overline{1,m}$, i.e. all fixed servers are busy and priorities do not work in this case. We have therefore

$$\min[K,N]\lambda\cdot p\{(0,0),(0,0),\ldots,(0,0)\}=$$

$$=\sum_{j=1}^{m}\mu_j p\{(0,0),\ldots,(0,0),(n_j=1,k_j=1),(0,0),\ldots,(0,0)\}, \tag{3}$$

for the state $\{(0,0),\ldots,(0,0)\}$, i.e. $n_i=k_i=0$, $i=\overline{1,m}$;

$$\left[\sum_{j=1}^{s}k_{i_j}\mu_{i_j}+\min[K,\left(N-\sum_{l=1}^{s}k_{i_l}\right)]\lambda\right]p\{(k_1,k_1),(k_2,k_2),\ldots,(k_m,k_m)\}=$$

$$=\sum_{j=1}^{m}(k_j+1)\mu_j p\{(k_1,k_1),\ldots,(k_{j-1},k_{j-1}),(k_j+1,k_j+1),(k_{j+1},k_{j+1}),\ldots,(k_m,k_m)\}+$$

$$+\sum_{j=1}^{s}\min[K,\left(N-\sum_{l=1}^{s}k_{i_l}+1\right)]\lambda p_{i_j}*$$

$$*\,p\{(k_1,k_1),\ldots,(k_{i_j-1},k_{i_j-1}),(k_{i_j}-1,k_{i_j}-1),(k_{i_j+1},k_{i_j+1}),\ldots,(k_m,k_m)\}, \tag{4}$$

for the states $n_i=k_i$, $i=\overline{1,m}$, $k_{i_j}>0$, $j=\overline{1,s}$, $k_{i_j}=0$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{j=1}^{s}n_{i_j}=\sum_{j=1}^{s}k_{i_j}<N$;

$$\left[\sum_{j=1}^{s}k_{i_j}\mu_{i_j}\right]p\{(k_1,k_1),(k_2,k_2),\ldots,(k_m,k_m)\}=$$

$$=\sum_{j=1}^{s}\lambda p_{i_j}p\{(k_1,k_1),\ldots,(k_{i_j-1},k_{i_j-1}),(k_{i_j}-1,k_{i_j}-1),(k_{i_j+1},k_{i_j+1}),\ldots,(k_m,k_m)\}, \tag{5}$$

for the states $n_i = k_i$, $i = \overline{1, m}$, $k_{i_j} > 0$, $j = \overline{1, s}$, $k_{i_j} = 0$, $j = \overline{s+1, m}$, $0 < s < m$, $\sum_{j=1}^{s} n_{i_j} = \sum_{j=1}^{s} k_{i_j} = N$;

$$\left[\sum_{i=1}^{m} k_i \mu_i + \min[K, \left(N - \sum_{j=1}^{m} k_j\right)]\lambda\right] p\{(k_1, k_1), (k_2, k_2), \dots, (k_m, k_m)\} =$$

$$= \sum_{j=1}^{m} (k_j + 1)\mu_j p\{(k_1, k_1), \dots, (k_{j-1}, k_{j-1}), (k_j + 1, k_j + 1), (k_{j+1}, k_{j+1}), \dots, (k_m, k_m)\} +$$

$$+ \sum_{j=1}^{m} \min[K, \left(N - \sum_{i=1}^{m} k_i + 1\right)]\lambda p_j *$$

$$* \, p\{(k_1, k_1), \dots, (k_{j-1}, k_{j-1}), (k_j - 1, k_j - 1), (k_{j+1}, k_{j+1}), \dots, (k_m, k_m)\}, \tag{6}$$

for the states $n_i = k_i > 0$, $i = \overline{1, m}$, $\sum_{i=1}^{m} n_i = \sum_{i=1}^{m} k_i < N$;

$$\left[\sum_{i=1}^{m} k_i \mu_i\right] p\{(k_1, k_1), (k_2, k_2), \dots, (k_m, k_m)\} =$$

$$= \sum_{j=1}^{m} \lambda p_j p\{(k_1, k_1), \dots, (k_{j-1}, k_{j-1}), (k_j - 1, k_j - 1), (k_{j+1}, k_{j+1}), \dots, (k_m, k_m)\}, \tag{7}$$

for the states $n_i = k_i > 0$, $i = \overline{1, m}$, $\sum_{i=1}^{m} n_i = \sum_{i=1}^{m} k_i = N$;

Now equations (1) follow from equations (3)-(7).

2) Next we consider the case $r < N$. Now priorities are working. Then we have

equation (3) again for the state $\{(0,0), \dots, (0,0)\}$, i.e. $n_i = k_i = 0$, $i = \overline{1, m}$;

equation (4) for the states $n_i = k_i$, $i = \overline{1, m}$, $k_{i_j} > 0$, $j = \overline{1, s}$, $k_{i_j} = 0$, $j = \overline{s+1, m}$,

$0 < s < m$, $\sum_{j=1}^{s} n_{i_j} = \sum_{j=1}^{s} k_{i_j} < r$;

equation (6) for the states $n_i = k_i > 0$, $i = \overline{1, m}$, $\sum_{i=1}^{m} n_i = \sum_{i=1}^{m} k_i < r$;

Other equations will differ.

$$\left[\sum_{j=1}^{s} k_{i_j} \mu_{i_j} + \min[K, (N - r)]\lambda\right] p\{(k_1, k_1), (k_2, k_2), \dots, (k_m, k_m)\} =$$

$$= \sum_{j=1}^{s} k_{i_j} \mu_{i_j} p\{(k_1, k_1), \ldots, (k_{i_j-1}, k_{i_j-1}), (k_{i_j}+1, k_{i_j}), (k_{i_j+1}, k_{i_j+1}), \ldots, (k_m, k_m)\} +$$

$$+ \sum_{j=1}^{m} \sum_{\substack{l=1 \\ i_l \neq j}}^{s} (k_j + 1)\mu_j p\{(k_1, k_1), \ldots, (k_{i_l-1}, k_{i_l-1}), (k_{i_l}, k_{i_l} - 1), (k_{i_l+1}, k_{i_l+1}), \ldots, (k_{j-1}, k_{j-1}),$$

$$* p\{(k_1, k_1), \ldots, (k_{i-1}, k_{i-1}), (k_i + \delta_{ij}, k_i - 1 + \delta_{ij}), (k_{i+1}, k_{i+1}), \ldots, (k_{j-1}, k_{j-1}),$$

$$(k_j + 1, k_j + 1 - \delta_{ij}), (k_{j+1}, k_{j+1}), \ldots, (k_m, k_m)\} +$$

$$(k_j + 1, k_j + 1), (k_{j+1}, k_{j+1}), \ldots, (k_m, k_m)\} +$$

$$+ \sum_{j=1}^{s} \min[K, (N - r + 1)]\lambda p_{i_j} *$$

$$* p\{(k_1, k_1), \ldots, (k_{i_j-1}, k_{i_j-1}), (k_{i_j} - 1, k_{i_j} - 1), (k_{i_j+1}, k_{i_j+1}), \ldots, (k_m, k_m)\}, \qquad (8)$$

for the states $n_i = k_i$, $i = \overline{1, m}$, $k_{i_j} > 0$, $j = \overline{1, s}$, $k_{i_j} = 0$, $j = \overline{s+1, m}$, $0 < s < m$, $\sum_{j=1}^{s} n_{i_j} = \sum_{j=1}^{s} k_{i_j} = r$;

$$\left[\sum_{i=1}^{m} k_i \mu_i + \min[K, (N - r)]\lambda\right] p\{(k_1, k_1), (k_2, k_2), \ldots, (k_m, k_m)\} =$$

$$= \sum_{j=1}^{m} \sum_{i=1}^{m} (k_j + 1 - \delta_{ij})\mu_j * + \sum_{j=1}^{m} (N - r + 1)\lambda p_j *$$

$$* p\{(k_1, k_1), \ldots, (k_{j-1}, k_{j-1}), (k_j - 1, k_j - 1), (k_{j+1}, k_{j+1}), \ldots, (k_m, k_m)\}, \qquad (9)$$

for the states $n_i = k_i > 0$, $i = \overline{1, m}$, $\sum_{i=1}^{m} n_i = \sum_{i=1}^{m} k_i = r$;

$$\left[\sum_{i=1}^{m} k_i \mu_i + \min[K, \left(N - \sum_{j=1}^{m} n_j\right)]\lambda\right] p\{(n_1, k_1), (n_2, k_2), \ldots, (n_m, k_m)\} =$$

$$= \sum_{j=1}^{m} k_j \mu_j \left(1 - \min\left(\sum_{s=1}^{j-1} (n_s - k_s), 1\right)\right) *$$

$$* p\{(n_1, k_1), \ldots, (n_{j-1}, k_{j-1}), (n_j + 1, k_j), (n_{j+1}, k_{j+1}), \ldots, (n_m, k_m)\} +$$

$$+\sum_{j=1}^{m}\sum_{\substack{l=1\\ i_l\neq j}}^{t}(k_j+1)\mu_j\left(1-\min\left(\sum_{s=1}^{i_l-1}(n_s-k_s),1\right)\right)*$$

$$*\,p\{(n_1,k_1),\dots,(n_{i_l-1},k_{i_l-1}),(n_{i_l},k_{i_l}-1),(n_{i_l+1},k_{i_l+1}),\dots,(n_{j-1},k_{j-1}),$$

$$(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\dots,(n_m,k_m)\}+$$

$$+\sum_{j=1}^{s}\min[K,\left(N-\sum_{i=1}^{m}n_i+1\right)]\lambda p_{i_j}\,*$$

$$*\,p\{(n_1,k_1),\dots,(n_{i_j-1},k_{i_j-1}),(n_{i_j}-1,k_{i_j}),(n_{i_j+1},k_{i_j+1}),\dots,(n_m,k_m)\}, \qquad (10)$$

for the states $k_{i_l}>0$, $l=\overline{1,t}$, $k_{i_l}=0$, $l=\overline{t+1,m}$, $0<t<m$, $n_{i_j}>k_{i_j}$, $j=\overline{1,s}$, $n_{i_j}=k_{i_j}$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{l=1}^{t}k_{i_l}=r$, $\sum_{i=1}^{m}n_i<N$;

$$\left[\sum_{i=1}^{m}k_i\mu_i+\min[K,\left(N-\sum_{j=1}^{m}n_j\right)]\lambda\right]p\{(n_1,k_1),(n_2,k_2),\dots,(n_m,k_m)\}=$$

$$=\sum_{j=1}^{m}k_j\mu_j\left(1-\min\left(\sum_{s=1}^{j-1}(n_s-k_s),1\right)\right)*$$

$$*\,p\{(n_1,k_1),\dots,(n_{j-1},k_{j-1}),(n_j+1,k_j),(n_{j+1},k_{j+1}),\dots,(n_m,k_m)\}+$$

$$+\sum_{j=1}^{m}\sum_{\substack{i=1\\ i\neq j}}^{m}(k_j+1)\mu_j\left(1-\min\left(\sum_{s=1}^{i-1}(n_s-k_s),1\right)\right)*$$

$$*\,p\{(n_1,k_1),\dots,(n_{i-1},k_{i-1}),(n_i,k_i-1),(n_{i+1},k_{i+1}),\dots,(n_{j-1},k_{j-1}),$$

$$(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\dots,(n_m,k_m)\}+$$

$$+\sum_{j=1}^{s}\min[K,\left(N-\sum_{i=1}^{m}n_i+1\right)]\lambda p_{i_j}\,*$$

$$*\,p\{(n_1,k_1),\dots,(n_{i_j-1},k_{i_j-1}),(n_{i_j}-1,k_{i_j}),(n_{i_j+1},k_{i_j+1}),\dots,(n_m,k_m)\}, \qquad (11)$$

for the states $k_i>0$, $i=\overline{1,m}$, $n_{i_j}>k_{i_j}$, $j=\overline{1,s}$, $n_{i_j}=k_{i_j}$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{i=1}^{m}k_i=r$, $\sum_{i=1}^{m}n_i<N$;

$$\left[\sum_{i=1}^{m} k_i\mu_i + \min[K,\left(N-\sum_{j=1}^{m} n_j\right)]\lambda\right] p\{(n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m)\} =$$

$$= \sum_{j=1}^{m} k_j\mu_j \left(1-\min\left(\sum_{s=1}^{j-1}(n_s-k_s),1\right)\right) *$$

$$* \; p\{(n_1,k_1),\ldots,(n_{j-1},k_{j-1}),(n_j+1,k_j),(n_{j+1},k_{j+1}),\ldots,(n_m,k_m)\} +$$

$$+ \sum_{j=1}^{m}\sum_{\substack{l=1\\ i_l\neq j}}^{t}(k_j+1)\mu_j\left(1-\min\left(\sum_{s=1}^{i_l-1}(n_s-k_s),1\right)\right) *$$

$$* \; p\{(n_1,k_1),\ldots,(n_{i_l-1},k_{i_l-1}),(n_{i_l},k_{i_l}-1),(n_{i_l+1},k_{i_l+1}),\ldots,(n_{j-1},k_{j-1}),$$

$$(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\ldots,(n_m,k_m)\} +$$

$$+ \sum_{j=1}^{m} \min[K,\left(N-\sum_{i=1}^{m} n_i+1\right)]\lambda p_j \, *$$

$$* \; p\{(n_1,k_1),\ldots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\ldots,(n_m,k_m)\}, \qquad (12)$$

for the states $k_i < n_i$, $i=\overline{1,m}$, $k_{i_l} > 0$, $l=\overline{1,t}$, $k_{i_l}=0$, $l=\overline{t+1,m}$, $0<t<m$, $\sum_{l=1}^{t} k_{i_l} = r$, $\sum_{i=1}^{m} n_i < N$;

$$\left[\sum_{i=1}^{m} k_i\mu_i + \min[K,\left(N-\sum_{j=1}^{m} n_j\right)]\lambda\right] p\{(n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m)\} =$$

$$= \sum_{j=1}^{m}\sum_{i=1}^{m}(k_j+1-\delta_{ij})\mu_j\left(1-\min\left(\sum_{s=1}^{i-1}(n_s-k_s),1\right)\right) *$$

$$* \; p\{(n_1,k_1),\ldots,(n_{i-1},k_{i-1}),(n_i+\delta_{ij},k_i-1+\delta_{ij}),(n_{i+1},k_{i+1}),\ldots,(n_{j-1},k_{j-1}),$$

$$(n_j+1,k_j+1-\delta_{ij}),(n_{j+1},k_{j+1}),\ldots,(n_m,k_m)\} +$$

$$+ \sum_{j=1}^{m}[K,\left(N-\sum_{i=1}^{m} n_i+1\right)]\lambda p_j \, *$$

$$* \; p\{(n_1,k_1),\ldots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\ldots,(n_m,k_m)\}, \qquad (13)$$

for the states $0<k_i<n_i$, $i=\overline{1,m}$, $\sum_{i=1}^{m} k_i = r$, $\sum_{i=1}^{m} n_i < N$;

$$\left[\sum_{i=1}^{m} k_i \mu_i\right] p\{(n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m)\} =$$

$$= \sum_{j=1}^{s} \lambda p_{i_j}\, p\left\{(n_1,k_1),\ldots,\left(n_{i_j-1},k_{i_j-1}\right),\left(n_{i_j}-1,k_{i_j}\right),\left(n_{i_j+1},k_{i_j+1}\right),\ldots,(n_m,k_m)\right\}, \qquad (14)$$

for the states $k_i \geq 0$, $i=\overline{1,m}$, $n_{i_j} > k_{i_j}$, $j=\overline{1,s}$, $n_{i_j} = k_{i_j}$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{i=1}^{m} k_i = r$, $\sum_{i=1}^{m} n_i = N$;

$$\left[\sum_{i=1}^{m} k_i \mu_i\right] p\{(n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m)\} =$$

$$= \sum_{j=1}^{m} \lambda p_j\, p\{(n_1,k_1),\ldots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\ldots,(n_m,k_m)\}, \qquad (15)$$

for the states $0 \leq k_i < n_i$, $i=\overline{1,m}$, $\sum_{i=1}^{m} k_i = r$, $\sum_{i=1}^{m} n_i = N$;

Now equations (1) follow from equations (3), (4), (6), (8)-(15).

Q.E.D.

IV. Performance Characteristics

In this Section we shall show how to compute some useful performance characteristics of the RTS under consideration.
Each server can be in one of following positions:
(i) busy (operating);
(ii) out of order (in maintenance);
(iii) on stand-by;
(iv) waiting for maintenance.
Each channel can be in one of two positions:
(i) in service;
(ii) out of service.

Keeping in mind that, the state of the system is represented by the vector $\{(n_1,k_1),\ldots,(n_m,k_m)\}$, where n_i ($i=1,\ldots,m$) is the total number of fixed servers of i-th activity type and k_i ($i=1,\ldots,m$) is the number of operating fixed servers of i-th activity type, we can represent the number of servers and channels in different positions in terms of n_i and k_i, namely:

Number of fixed servers is $\sum_{i=1}^{m} n_i$;

Number of operating fixed servers (the number of channels in service) is $\sum_{i=1}^{m} k_i$;

Number of fixed servers of i-th activity type being on stand-by is $n_i - k_i$;

Number of all fixed servers being on stand-by is $\sum_{i=1}^{m}(n_i - k_i)$;

Number of broken servers is $N - \sum_{i=1}^{m} n_i$;

Number of channels out of service is $r - \sum_{i=1}^{m} k_i$;

Number of broken servers in maintenance is $\min\left(N - \sum_{i=1}^{m} n_i, K\right)$;

Number of broken servers waiting for maintenance is $\max\left(N - \sum_{i=1}^{m} n_i - K, 0\right)$.

Now we can obtain corresponding average values, using steady state probabilities, namely:

$L_i = E[n_i]$ - average number of fixed servers of *i*-th activity type ($i = \overline{1, m}$);

$L = \sum_{i=1}^{m} L_i$ - average number of all fixed servers;

$N - L$ - average number of broken servers;

$E[k_i]$ - average number of operating servers of *i*-th activity type ($i = \overline{1, m}$);

$\sum_{i=1}^{m} E[k_i]$ - average number of all operating servers (as well as the average number of channels in service);

$r - \sum_{i=1}^{m} E[k_i]$ - average number of non-executed real-time jobs;

$$L_m = E\left[\min\left(N - \sum_{i=1}^{m} n_i, K\right)\right]$$

for average number of broken servers in maintenance;

$$L_w = E\left[\max\left(N - \sum_{i=1}^{m} n_i - K, 0\right)\right]$$

for average number of broken servers waiting for maintenance;

$L_{i,sb} = E[n_i - k_i]$ - average number of *i*-th type fixed servers on stand-by ($i = \overline{1, m}$);

$L_{sb} = \sum_{i=1}^{m} L_{i,sb}$ - average number of all fixed servers on stand-by;

$\lambda_{av} = \lambda(N - L)$ - average rate of servers arriving from maintenance;

$\lambda_{av,i} = \lambda p_i (N - L)$ - average rate of *i*-th type servers arriving from maintenance ($i = \overline{1, m}$);

$W_{i,sb} = L_{i,sb} / \lambda_{av,i}$ - average time spent on stand-by by the *i*-th type server ($i = \overline{1, m}$) - Little Theorem;

$W_i = W_{i,sb} + 1/\mu_i$ - average time of the *i*-th type server being fixed ($i = \overline{1, m}$) - Little Theorem.

The use of RTS relies on the principle of availability, which can be given in our case by the following formula

$$Av = \frac{1}{r}\sum_{i=1}^{m} E[k_i].$$

Another important performance measure is the average cost of system operation during time unit. Denote C_i (i=1,…,m) the cost of time unit activity of the i-th type server, and D the penalty paid for the time unit during which one of channels/jobs was not served. Then the average cost per time unit of system operation is defined as follows

$$TC = \sum_{i=1}^{m} C_i E[k_i] + D\left(r - \sum_{i=1}^{m} E[k_i]\right) = Dr - \sum_{i=1}^{m}(D - C_i)E[k_i]$$

V. Conclusions

In this paper we provided equations for steady-state probabilities, as well as formulas for availability, average cost function and other performance characteristics of multiserver and multichannel RTS with different activity types and shortage of maintenance teams. We have examined models with nonpreemptive priorities. Practitioners and researchers can submit their parameters in equations for steady state probabilities and performance characteristics and to obtain the solutions. Further research should address following problems:

- RTS with preemptive priorities;
- RTS in which each server needs more than one kind of maintenance;
- Optimization of RTS involving costs and penalties;
- Multilevel RTS;

The use of queueing theory methodology will have significant benefits in analysis of these systems.

References

[1] C.L. Liu and J.W Layland, Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *J.ACM*,20(1), 1973, 46-61.

[2] S. K. Dhal and C. L. Liu, On a real-time scheduling problem, *Operations Research*, **26**, no. 1, 1978,127-140.

[3] M. Xie et al., Optimal number of hosts in a distributed system based on cost criteria, *International Journal of Systems Science***,** 35(6), 2004, 343-353.

[4] F.W. Glover and M. Laguna, *Tabu Search* (Dordrecht, Kluwer Academic Publishers, 1997).

[5] T.A. Feo, K. Venkatraman and J.F. Bard, A GRASP for a difficult single scheduling problem, *Computers and Operations Research*, 18,1991, 635-643,

[6] R. Seguin, et. al., Real-time decision problems: An operational research perspective, *Journal of the Operational Research Society*, 48, 1997, 162-174.

[7] J.K. Strosnider and C.J. Paul, A structured view of real-time problem solving, *AI Magazine*, 14, 1993, 45-66.

[8] J.Kreimer and A. Mehrez, An optimal operation policy for real-time n-server stand-by system involving preventive maintenance, *European Journal of Operational Research*, 69**,** 1993, 50-54.

[9] J.Kreimer and A. Mehrez, Optimal Real-Time Data Acquisition and Processing by a Multiserver Stand-By System, *Operations Research*,42(1), 1994, 24-30.

[10] J.Kreimer and A. Mehrez, Computation of Availability of a Real-Time System Using Queueing Theory Methodology, *Journal of the Operational Research Society*,49, 1998, 1095-1100.

[11] J.Kreimer, Real-time multiserver and multichannel systems with shortage of maintenance crews, *Mathematical and Computer Modelling*,30(11-12), 1999, 169-176.

[12] E. Ianovsky andJ. Kreimer,Multiserver Multichannel Real-Time System with Limited Maintenance Facilities under Maximum Load, *Open Journal of Applied Sciences*, Nu 5, 2015, 368-375.

[13] D. Gross, J.F. Shortle, J.M. Thompson and C.M. Harris, *Fundamentals of Queueing Theory*, 4th edition, (New York, John Wiley, 2008).

[14] J. Kreimer, Real-Time Multiserver System with Two Non-identical Channels and Limited Maintenance Facilities, *Mathematics and Computers in Simulation*, 53(1-2), 2000, 85-94.

[15] E. Ianovsky andJ. Kreimer,Optimization of Real-Time Multiserver System with Two Different Channels and Shortage of Maintenance Teams, *Mathematics and Computers in Simulation*, 63(6), 2003, 615-627.

[16] J. Kreimer, Real-time system with homogeneous servers and nonidentical channels in steady state, *Computers &Operations Research*,**29**(11), 2002, 1465-1473.

[17] J. Kreimer, Effectiveness analysis of real-time data acquisition and processing multichannel systems, *IEEE J. on Reliability*, 51(1), 2002, 91-99.

[18] J. Kreimer,Performance Evaluation of Multiple Unmanned Aerial Vehicles Operating in General Regime with Shortage of Maintenance Facilities, *Journal of Computer and Communications*, Nu 3, 2016, 32-45.

[19] J. Kreimer, Multiserver Single-channel real-time system with different kinds of activities, *Communications in Dependability and Quality Management*,6(1), 2003, 91-100.

[20] E. Ianovsky andJ. Kreimer,AnOptimal Routing Policy for Unmanned Aerial Vehicles (Analytical and Cross-EntropySimulation Approach),*Annals of Operations Research* SI MCsimulation, 189(2), 2011, 215-253.

[21] R.Y. Rubinstein and D.P. Kroese, *Simulation and the Monte Carlo Method*, 2nd edition (New York, John Wiley & Sons, 2008).

[22] R. Y.Rubinstein, A. Ridder and R. Vaisman, *Fast Sequential Monte Carlo Methods for Counting and Optimization* (New York John Wiley & Sons, 2014).

[23] D.P. Kroese, T. Taimre and Z.I. Botev, *Handbook of Monte Carlo Methods*, New York, John Wiley & Sons, 2011).

[24] E. Bassan and J. Kreimer, Multiserver and Multichannel Real-Time Systems with Separate Queues and Preemptive priorities.*Computer Modelling and New Technologies*, 42(2), 2008, 7-15.

[25] E. Bassan and J. Kreimer, "Multichannel Real-Time System with Single Joint Queue and Preemptive Priorities", *Communications in Dependability and Quality Management*, 12(2), 2009, 37-50.

[26] E. Bassan and J. Kreimer, Analytical approximations of real-time systems with a single joint queue and preemptive prioritiesworking under maximal load, *InternationalJournal of Scientific and Technical Research in Engineering*, Nu 5, 2016, 38-48.

[27] J. KreimerandE. Ianovsky, Real Time Systems with Nonpreemptive Priorities and Ample Maintenance Facilities, *Journal of Computer and Communications*, Nu 3, 2015, 32-45.