PERGAMON Computers and Mathematics with Applications 38 (1999) 13–21

An International Journal
computers & mathematics with applications

www.elsevier.nl/locate/camwa

Optimization Problems on the Performance of a Nonreliable Terminal System

B. ALMÁSI AND J. SZTRIK
Institute of Mathematics and Informatics
University of Debrecen
P.O. Box 12, H-4010 Debrecen, Hungary
<almasi><jsztrik>@math.klte.hu

(Received and accepted March 1999)

Abstract—The aim of this paper is to investigate the effect of the different service disciplines, such as FIFO, PS, Priority Processor Sharing, Polling, on the main performance measures, such as utilizations, response times, throughput, mean queue length. It has been shown by numerical examples that even in the case of homogeneous sources and homogeneous failure and repair times, the CPU utilization depends on the scheduling discipline contrary to the case of reliable terminal systems. All random variables involved in the model construction are supposed to be exponentially distributed and independent of each other. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords—Nonreliable terminal system, Performance optimization, Scheduling rules, Service disciplines.

1. INTRODUCTION

The machine interference model (sometimes called machine-repairman model, finite-source model) has been treated in many forms over the past years. It has often been used in analysing multiterminal systems under different scheduling rules, cf. [1,2]. The optimal operation of finite-source systems has been one of the main objectives of recent research, see for example, [3–6].

In this paper, we consider a stochastic queueing model for the performance evaluation of a computer system consisting of n terminals connected with a CPU. A user at terminal i has thinking and processing times, respectively, depending on index i. Let us suppose that the operational system is subject to random breakdowns, which may be software and hardware ones, stopping the service both at the terminals and at the CPU. The failure-free operation times of the system and the restoration times are random variables. The busy terminals are also subject to random breakdowns not affecting the system's operation. The failure-free operation times and the repair times of busy terminal i are random variables with distribution function depending on index i. The breakdowns are serviced by a single repairman providing pre-emptive priority to the system's failure, while the restorations at the terminals are carried out according to FIFO rule.

This work was partially supported by the Hungarian National Foundation for Scientific Research under Grant OTKA T014974/95 and OTKA T016933/95.

0898-1221/99/$ - see front matter. © 1999 Elsevier Science Ltd. All rights reserved. Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TeX
PII: S0898-1221(99)00202-3

We assume that each user generates only one job at a time, and he waits at the CPU before he starts thinking again, that is, the terminal is inactive while waiting at the CPU, and it cannot break down. Its importance is due to the fact that it is the simplest closed queueing network consisting of two nodes only. For more complex investigation of networks, this simple model can give some insight into the effects of different system parameters, and in approximate analysis of large networks, they can be considered as building-blocks.

Several works have been devoted to the investigation of the utilization factor of the Central Processor Unit (CPU) and the number of jobs staying at the CPU. It has turned out that in the case when the involved random variables are exponentially distributed, the request's generation rates are the same, the processing rates are different [3,5], Lehtonen [7] and Van der Wal [8] have proved that the utilization of the CPU is not influenced at all by any work-conserving scheduling rule, including First-In-First-Out (FIFO), Processor Sharing (PS), Priority Processor Sharing (PPS), Pre-emptive or Nonpre-emptive priority, Shortest and Longest-Expected-Processing-Time-First disciplines. More precisely, it has been shown that the mean busy period length of the processor is the same for any of the above-mentioned schedulings. Furthermore, the mean number of jobs staying at the CPU is minimized by giving higher pre-emptive priority to a job with less mean job size (so-called H-schedule). Consequently, the overall utilization of the system, the sum of CPU and terminal utilizations, sometimes called as effective degree of multiprogramming, is maximized. Based on this fact, Kameda [9] has investigated more practical models of multiprogramming systems to estimate the maximum processing capacity of the system.

In the case when the request's generation rates are also different by using different methods, Koole and Vrijenhoek [6] and Van der Wal [8] have shown that if pre-emptions of the resume type are allowed, the CPU utilization is maximized by giving higher priority to the jobs of the faster thinking terminals irrespective of the expected job sizes. Results for the overall device utilizations have not been mentioned. However, in practice, we can see that the terminals and the CPU are not always available for service. These situations could be considered as breakdowns, so the analysis of nonreliable terminal systems seems to be also important. Assuming that the involved random variables are independent and exponentially distributed, different models have been discussed. The homogeneous case, i.e., when the thinking times, processing times, failure-free operation times, and the restoration times are the same for all terminals, has been dealt with in [10]. The heterogeneous models under PPS, Polling and FIFO rule have been treated in [11,12], the main performance measures have been obtained by numerical and simulation approach. The aim of this paper is a synthesis of earlier numerical results with the intention to investigate the effect of the different scheduling disciplines, such as FIFO, PS, PPS, Polling, on the main performance measures, such as utilizations, response times, mean queue length.

2. MODEL FORMULATION

Let us consider a computer system consisting of $n \geq 2$ terminals connected with a CPU. A user at the terminal i has thinking and processing times, respectively, depending on index i. Let us suppose, as it was mentioned in [7], that the operational system is subject to random breakdowns, which may be software and hardware ones, stopping the service both at the terminals and at the CPU. The failure-free operation times of the system and the restoration times are random variables. The busy terminals are also subject to random breakdowns not affecting the system's operation. The failure-free operation times and the repair times of terminal i are random variables with distribution function depending on index i. The breakdowns are serviced by a single repairman providing pre-emptive priority to the system's failure, while the restorations at the terminals are carried out according to FIFO rule. Each user is assumed to generate only one job at a time, and he waits at the CPU before he starts thinking again, that is, the terminal is inactive while waiting at the CPU, and it cannot break down. All random variables involved in the model construction are supposed to be exponentially distributed and independent of each other.

To deal with the mathematical model, we have to introduce the following random variables (stochastic processes):

$$
\begin{aligned}
X(t) &:= \begin{cases} 1, & \text{if the operating system fails at time } t, \\ 0, & \text{otherwise,} \end{cases} \\
Y(t) &:= \text{the number of failed terminals at time } t, \\
YI(t) &:= \text{the failed terminals' indices at time } t \text{ in order of their failure, or 0 if } Y(t) = 0, \\
Z(t) &:= \text{the number of jobs residing at the CPU at time } t, \\
ZI(t) &:= \text{the indices of these jobs.}
\end{aligned} \tag{1}
$$

Depending on the service discipline the random variable $ZI(t)$ gives the order of service by the CPU, too. It can easily be seen that, under the exponential distribution condition, the multidimensional stochastic process $M(t) = (X(t), Y(t), YI(t), Z(t), ZI(t))$ is a Markov chain having a rather complex, and large state space. To get its steady-state probabilities, an efficient recursive computational method has been introduced and used for different service rules mentioned earlier, cf. [10–12]. Let us denote the steady-state distribution of $(M(t), t \geq 0)$ by

$$
\begin{aligned}
P(q; i_1 \dots i_k; j_1, \dots, j_s) = \lim_{t\to\infty} P(X(t) = q;\ Y(t) = k; \\
YI(t) = i_1, \dots, i_k;\ Z(t) = s;\ ZI(t) = j_1, \dots, j_s).
\end{aligned} \tag{2}
$$

Furthermore, let us denote by $P(q, k, s)$ $(q = 0, 1; k = 1, \dots, n; s = 1, \dots, n-k)$ the steady-state probability that the operating system is in state q, k terminals are failed and s jobs are at the CPU. Assuming that these probabilities exist and are known, the main performance measures can be obtained as follows (see [11]).

(i) Mean number of jobs residing at the CPU

$$\overline{n}_j = \sum_{i=0}^{1} \sum_{k=0}^{n} \sum_{s=0}^{n-k} sP(i, k, s).$$

(ii) Mean number of working terminals

$$\overline{n}_g = n - \sum_{i=0}^{1} \sum_{k=0}^{n} \sum_{s=0}^{n-k} kP(i, k, s).$$

(iii) Average number of busy terminals

$$\overline{n}_b = \sum_{k=0}^{n} \sum_{s=0}^{n-k} (n-k-s)P(0, k, s).$$

(iv) Utilization of the repairman

$$U_r = \sum_{k=0}^{n} \sum_{s=0}^{n-k} P(1, k, s) + \sum_{k=1}^{n} \sum_{s=0}^{n-k} P(0, k, s).$$

(v) Utilization of the CPU

$$U_{\text{CPU}} = \sum_{k=0}^{n-1} \sum_{s=1}^{n-k} P(0, k, s).$$

(vi) Utilization of the i^{th} terminal, $i = 1, \ldots, n$

$$U_i = \sum_{k=0}^{n} \sum_{s=0}^{n-k} \sum_{r=1}^{k} \sum_{v=1}^{s} \sum_{i_1,\ldots,i_k} \sum_{j_1,\ldots,j_s} (1 - \delta(i, i_r) - \delta(i, j_v)) P(0; i_1, \ldots, i_k; j_1, \ldots, j_s),$$

where $\delta(i,j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$

(vii) Overall utilization of the system

$$U = \sum_{i=1}^{n} U_i + U_{\mathrm{CPU}} + U_r.$$

(viii) Expected response time of jobs for terminal i

$$T_i = \frac{Q_i}{\lambda_i U_i},$$

Q_i denotes the probability of staying at the CPU for the i^{th} terminal, namely,

$$Q_i = \sum_{q=0}^{1} \sum_{k=0}^{n-1} \sum_{s=1}^{n-k} \sum_{r=1}^{s} \sum_{i_1,\ldots,i_k} \sum_{j_1,\ldots,j_s} \delta(i, j_r) P(q; i_1, \ldots, i_k; j_1, \ldots, j_s).$$

It is easy to see that $n_b = \sum_{i=1}^{n} U_i$. Furthermore, let us denote by T the overall response time of the system, defined by $T := \sum_{i=1}^{n} T_i$, which is also a very important measure of effectiveness.

3. COMPUTATIONAL RESULTS AND THEIR EXPLANATION

In this section, we give several numerical examples to illustrate the effect of different system parameters on the performance measures calculated on the basis of (2) for $n = 4, 5$ and (i)–(viii). As it is well known (see, e.g., [1,2]), that the Pre-emptive Priority discipline can be approximated by the PPS rule by assigning appropriate weights to the corresponding jobs; that is the reason why it will not be mentioned separately.

Let us denote by λ_i, μ_i, γ_i, τ_i, w_i the parameters of the exponentially distributed thinking, processing, operating, repair times and weight for terminal i, $i = 1, \ldots, n$, respectively. Similarly, let α, β denote the failure and repair rate of the CPU, respectively.

CASE 1. Input parameters:

$n = 4$	$\alpha = 0.001$	$\beta = 999.0$

i	λ_i	μ_i	γ_i	τ_i	w_i
1	0.3500	0.4000	0.2000	0.3000	3.0
2	0.3500	0.8500	0.2000	0.3000	90.0
3	0.3500	0.5000	0.2000	0.3000	15.0
4	0.3500	0.9000	0.2000	0.3000	190.0

Table 1. Performance measures.

	FIFO	PS	POLLING	PPS
n_b	1.1313	1.1519	1.1310	1.1865
U_r	0.7542	0.7680	0.7540	0.7910
U_{CPU}	0.6631	0.6605	0.6631	0.6559
U	2.5486	2.5804	2.5481	2.6334
U_1	0.2683	0.2500	0.2685	0.2117
U_2	0.2926	0.3140	0.2926	0.3454
U_3	0.2764	0.2695	0.2765	0.2688
U_4	0.2941	0.3185	0.2933	0.3606
T_1	3.7852	4.3980	3.7767	6.0742
T_2	2.9093	2.3223	2.9125	1.5663
T_3	3.4755	3.6504	3.4699	3.5635
T_4	2.8605	2.2101	2.8826	1.2886
T	13.031	12.581	13.042	12.497

CASE 2. Input parameters:

$n = 5$	$\alpha = 0.001$	$\beta = 999.0$

i	λ_i	μ_i	γ_i	τ_i	w_i
1	0.3500	0.4000	0.2000	0.3000	3.0
2	0.3500	0.8500	0.2000	0.3000	90.0
3	0.3500	0.5000	0.2000	0.3000	15.0
4	0.3500	0.9000	0.2000	0.3000	190.0
5	0.3500	0.6000	0.2000	0.3000	40.0

Table 2. Performance measures.

	FIFO	PS	POLLING	PPS
n_b	1.2271	1.2466	1.2267	1.2841
U_r	0.8181	0.8311	0.8178	0.8561
U_{CPU}	0.7195	0.7163	0.7195	0.7095
U	2.7647	2.7947	2.7640	2.8497
U_1	0.2344	0.2154	0.2349	0.1694
U_2	0.2529	0.2734	0.2529	0.3055
U_3	0.2406	0.2330	0.2410	0.2236
U_4	0.2540	0.2776	0.2535	0.3224
U_5	0.2452	0.2471	0.2445	0.2634
T_1	4.4009	5.2237	4.3834	8.1091
T_2	3.5028	2.7596	3.5088	1.7521
T_3	4.0826	4.3359	4.0690	4.5728
T_4	3.4530	2.6264	3.4724	1.3788
T_5	3.8575	3.7178	3.8803	2.9726
T	19.2968	18.6634	19.3139	18.7854

CASE 3. Input parameters:

$n = 4$	$\alpha = 0.05$	$\beta = 1.0$

i	λ_i	μ_i	γ_i	τ_i	w_i
1	0.3500	0.4000	0.2000	0.3000	3.0
2	0.3500	0.8500	0.2000	0.3000	90.0
3	0.3500	0.5000	0.2000	0.3000	15.0
4	0.3500	0.9000	0.2000	0.3000	190.0

Table 3. Performance measures.

	FIFO	PS	POLLING	PPS
n_b	1.0775	1.0971	1.0772	1.1300
U_r	0.7651	0.7782	0.7649	0.8002
U_{CPU}	0.6315	0.6291	0.6315	0.6247
U	2.4741	2.5044	2.4736	2.5549
U_1	0.2555	0.2381	0.2558	0.2016
U_2	0.2787	0.2990	0.2787	0.3289
U_3	0.2632	0.2567	0.2634	0.2560
U_4	0.2801	0.3033	0.2793	0.3434
T_1	3.9744	4.6179	3.9655	6.3779
T_2	3.0547	2.4384	3.0581	1.6446
T_3	3.6493	3.8329	3.6433	3.7416
T_4	3.0035	2.3205	3.0267	1.3530
T	13.6819	13.2097	13.6936	13.1171

CASE 4. Input parameters:

$n = 4$	$\alpha = 0.001$	$\beta = 999.0$

i	λ_i	μ_i	γ_i	τ_i	$w_i - 1$	$w_i - 2$	$w_i - 3$
1	0.1000	0.6000	0.2000	0.3000	10.0	30.0	1.0
2	0.2000	0.8000	0.2000	0.3000	30.0	20.0	10.0
3	0.3000	0.7000	0.2000	0.3000	20.0	10.0	20.0
4	0.4000	0.5000	0.2000	0.3000	1.0	1.0	30.0

Table 4. Performance measures.

	FIFO	PS	POLLING	PPS-1	PPS-2	PPS-3
n_b	1.2787	1.2833	1.2790	1.2969	1.2980	1.2677
U_r	0.8525	0.8556	0.8527	0.8646	0.8653	0.8451
U_{CPU}	0.4951	0.4934	0.4948	0.4875	0.4864	0.5018
U	2.6263	2.6323	2.6265	2.6490	2.6497	2.6146
U_1	0.3626	0.3618	0.3635	0.3648	0.3741	0.3311
U_2	0.3352	0.3428	0.3354	0.3585	0.3563	0.3357
U_3	0.3067	0.3113	0.3060	0.3277	0.3209	0.3177
U_4	0.2743	0.2675	0.2741	0.2458	0.2468	0.2832
T_1	2.8595	2.8577	2.8071	2.5190	2.0112	4.9519
T_2	2.3393	2.0629	2.3278	1.4995	1.5575	2.3805
T_3	2.3536	2.2112	2.3700	1.7366	1.8965	2.1025
T_4	2.6458	2.8199	2.6501	3.4467	3.4106	2.4449
T	10.198	9.9517	10.155	9.2018	8.8758	11.879

CASE 5. Input parameters:

$n = 4$	$\alpha = 0.01$	$\beta = 1.0$

i	λ_i	μ_i	γ_i	τ_i	$w_i - 1$	$w_i - 2$	$w_i - 3$
1	0.1000	0.6000	0.1000	0.3000	10.0	30.0	1.0
2	0.2000	0.8000	0.1500	0.3000	30.0	20.0	10.0
3	0.3000	0.7000	0.2000	0.3000	20.0	10.0	20.0
4	0.4000	0.5000	0.2500	0.3000	1.0	1.0	30.0

Table 5. Performance measures.

	FIFO	PS	POLLING	PPS-1	PPS-2	PPS-3
n_b	1.4006	1.4077	1.4013	1.4323	1.4371	1.3762
U_r	0.8270	0.8277	0.8269	0.8293	0.8283	0.8287
U_{CPU}	0.4858	0.4854	0.4857	0.4837	0.4827	0.4893
U	2.7134	2.7208	2.7139	2.7453	2.7481	2.6942
U_1	0.4851	0.4842	0.4863	0.4899	0.5053	0.4412
U_2	0.3845	0.3935	0.3848	0.4143	0.4110	0.3855
U_3	0.3124	0.3168	0.3117	0.3339	0.3257	0.3238
U_4	0.2187	0.2132	0.2186	0.1943	0.1952	0.2258
T_1	2.7512	2.7765	2.7097	2.5547	2.0409	4.4758
T_2	2.2915	2.0474	2.2846	1.5303	1.6095	2.2520
T_3	2.3559	2.2360	2.3733	1.8064	2.0054	2.0506
T_4	2.7918	3.0099	2.7972	3.8600	3.8190	2.5151
T	10.190	10.070	10.165	9.7514	9.4748	11.293

CASE 6. Input parameters:

$n = 4$	$\alpha = 0.01$	$\beta = 1.0$

i	λ_i	μ_i	γ_i	τ_i	$w_i - 1$	$w_i - 2$	$w_i - 3$
1	0.1000	0.6000	0.1000	0.2000	10.0	30.0	1.0
2	0.2000	0.8000	0.1500	0.3500	30.0	20.0	10.0
3	0.3000	0.7000	0.2000	0.3000	20.0	10.0	20.0
4	0.4000	0.5000	0.2500	0.3500	1.0	1.0	30.0

Table 6. Performance measures.

	FIFO	PS	POLLING	PPS-1	PPS-2	PPS-3
n_b	1.3715	1.3785	1.3720	1.4006	1.4026	1.3552
U_r	0.8292	0.8300	0.8292	0.8326	0.8328	0.8273
U_{CPU}	0.4807	0.4803	0.4806	0.4777	0.4759	0.4870
U	2.6814	2.6888	2.6818	2.7109	2.7113	2.6695
U_1	0.4580	0.4568	0.4589	0.4617	0.4751	0.4177
U_2	0.3855	0.3948	0.3859	0.4152	0.4114	0.3880
U_3	0.3080	0.3123	0.3074	0.3286	0.3202	0.3206
U_4	0.2200	0.2145	0.2199	0.1952	0.1959	0.2287
T_1	2.7560	2.7964	2.7173	2.5640	2.0438	4.5804
T_2	2.2953	2.0468	2.2859	1.5270	1.6015	2.2690
T_3	2.3594	2.2396	2.3760	1.8054	1.9980	2.0657
T_4	2.7919	3.0050	2.7973	3.8420	3.7964	2.5198
T	10.202	10.087	10.176	9.7384	9.4379	11.435

In Case 1, despite homogeneous thinking times, our calculations give that, contrary to the statement of Kameda [9], U_{CPU}s are different. It is the least for PPS; however, n_b and U are the greatest and T is the least under this scheduling, as it was expected.

In Case 2, we tried to show the effect of the number of terminals on the performance measures. It can be seen that T increased the values of Case 1 with 50 percent under each discipline.

In Case 3, we have the same parameters as earlier, except the CPU failure and repair (α, β). U, n_b, U_i decreased and T increased as it was expected.

In Case 4, the thinking and processing times are heterogeneous, the operating and repairing times are homogeneous. Three priority orderings have been considered. When the priority assignment takes place with respect to the decreasing order of thinking rates, the U_{CPU} is the highest as it was stated in [6,8]. In this case, the importance of the objective performance measure (U, T) should be underlined. It can be seen that for U_{CPU}, the PPS-3, and for U and T the PPS-2 discipline is optimal. At the same time, we can see that PPS-2 is a mixed priority assignment.

In Case 5, the CPU is subject to breakdowns, the failure rates (γ_i) are different and have the same arithmetic mean as in Case 4; the other system parameters are unchanged. U has increased under each discipline, T has decreased in FIFO and PPS-3, and it has increased in other cases.

In Case 6, the repair rates are different with the same arithmetic mean as in Case 5; the other system parameters have not been varied. U has decreased under each scheduling, T has increased in FIFO, PS, Polling and PPS-3 cases.

It is shown by numerical calculations that the effects of different system parameters and scheduling disciplines are unpredictable in many cases. For relatively small number of terminals, the performance measures can be calculated numerically. For greater values, only stochastic simulation is recommended.

4. CONCLUSIONS

A queueing model has been constructed for the mathematical description of a heterogeneous multiterminal system in which the CPU and the terminals are subject to random breakdowns. We can see that the most complicated case is pre-emptive priority scheduling since we do not know which parameters determine the priority assignment. So altogether, in principle, the number of possible cases is $4n!$, namely, assignment according to thinking, processing, operating and repair times. To reduce the number of cases, we suggest applying only FIFO, PS, Polling scheduling because the main performance measures are very close to the arithmetic mean of the different PPS runs, respectively. Finally, the importance of the objective performance measure should be emphasized, since even in social optimization it could easily occur that if a scheduling is optimal for a given measure, it will not be optimal for another one, cf. Case 4.

REFERENCES

1. L. Kleinrock, *Queueing Systems. Vol. I: Theory. Vol. II: Computer Applications*, Wiley-Interscience, New York, (1975).
2. H. Takagi, *Queueing Analysis. A Foundation of Performance Evaluation. Vol. 2: Finite Systems*, North-Holland, (1993).
3. D. Asztalos, Optimal control of finite-source priority queues with computer system applications, *Computers Math. Applic.* **6** (4), 425–431 (1980).
4. L.C. Goheen, On the optimal operating policy for the machine repair problem when failure and repair times have Erlang distribution, *Operations Research* **25**, 484–491 (1977).
5. H. Kameda, A finite-source queue with different customers, *J. ACM* **29**, 478–491 (1982).
6. G. Koole and M. Vrijenhoek, Scheduling a repairman in a finite source system, *Mathematical Methods of Operations Research* **44**, 333–344 (1996).
7. T. Lehtonen, On the optimal policies of an exponential machine repair problem, *Naval Research Logistics Quarterly* **31**, 173–181 (1984).
8. J. Van der Wal, The maximization of CP utilization in an exponential CP-terminal system with different think times and different job sizes, *Stochastic Processes and Their Applications* **18**, 277–289 (1984).
9. H. Kameda, The effect of CPU scheduling and job loading policies on the processing capacity of computer systems, *Stochastic Analysis of Computer and Communication Systems*, (Edited by H. Takagi), pp. 531–548, North Holland, (1990).
10. J. Sztrik and T. Gaál, A recursive solution of a queueing model for a multi-terminal system subject to breakdowns, *Performance Evaluation* **11**, 1–7 (1990).
11. B. Almási, Response time for finite heterogeneous nonreliable queueing systems, *Computers Math. Applic.* **31** (11), 55–59 (1996).
12. B. Almási, A queueing model for a nonhomogeneous polling system subject to breakdowns, *Annales Univ. Sci. Budapest. Sec. Comp.* (to appear).