DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

ONDERZOEKSRAPPORT NR 9632

ON CLOSED QUEUEING NETWORKS WITH MIXED PREEMPTIVE RESUME PRIORITY SERVERS

by

Herlinde Leemans

Guido Dedene

Katholieke Universiteit Leuven

Naamsestraat 69, B-3000 Leuven

ONDERZOEKSRAPPORT NR 9632

ON CLOSED QUEUEING NETWORKS WITH MIXED PREEMPTIVE RESUME PRIORITY SERVERS

by

Herlinde Leemans

Guido Dedene

D/1996/2376/32

On closed queueing networks with mixed preemptive resume priority servers

Herlinde Leemans
Katholieke Universiteit Leuven
Faculty of Economics and Applied Economics
Naamsestraat 69
B-3000 Leuven
Belgium

Fax : +32 -16- 32.67.32
E-mail : Herlinde.Leemans@econ.kuleuven.ac.be

Guido Dedene
Katholieke Universiteit Leuven
Faculty of Economics and Applied Economics
Naamsestraat 69
B-3000 Leuven
Belgium

Fax : +32 -16- 32.67.32
E-mail : Guido.Dedene@econ.kuleuven.ac.be

Abstract :

This paper discusses a typical closed queueing network model in which multiple preemptive resume servers are present with different priority structures at each priority node. An algorithm is developed that is applicable for the three-node two-class model and results are compared to point estimates obtained from simulation. The algorithm is partly based on the Delay/MVA algorithm developed by Bondi and Chuang, because of the accuracy with which instant arrival queue lengths at fcfs servers are calculated. Results are also compared with results obtained from the Shadow Approximation.

On closed queueing networks with mixed preemptive resume priority servers

Abstract :

This paper discusses a typical closed queueing network model in which multiple preemptive resume servers are present with different priority structures at each priority node. An algorithm is developed that is applicable for the three-node two-class model and results are compared to point estimates obtained from simulation. The algorithm is partly based on the Delay/MVA algorithm developed by Bondi and Chuang, because of the accuracy with which instant arrival queue lengths at fcfs servers are calculated. Results are also compared with results obtained from the Shadow Approximation.

0. Introduction

Queueing networks are widely used in computer performance modelling. The most general solution method for such networks is global balance equation solving. For networks of reasonable size (even with only a few service centers and a few customer classes), this may result in one global balance equation per feasible state of the network, requiring an enormous effort to solve. A broad class of queueing networks however, the BCMP networks [Baske75], have characteristics that allow a more efficient solution approach. They obey local balance and exhibit product form solutions. For closed product form queueing networks, approximation algorithms such as MVA [Reise75] and Convolution [Buzen73] have been developed that give reasonably accurate solutions to the performance measures.

BCMP type of networks are allowed to have different classes of customers with distinct routing and, under certain conditions, with distinct service time distributions. However, different priorities may not be assigned to these customer classes if the network is to retain a product form solution.

Many real-life systems do have different priorities for the different customer classes. This may have various reasons, e.g. application of the SPT-rule for minimal average response times and maximal throughput. Only few exact results for these models are available, since they can only be obtained by solving the (often extensive and complex) set of global balance equations. Avi-Itzhak and Heyman [Avi-I73] derive exact results for homogeneous closed queueing networks with preemptive-resume priorities observed on all servers. Morris [Morri81] extends these results for non-homogeneous two-node (tandem) models with preemptive-resume priorities on both nodes. The relaxation of the homogeneity assumption allows only two-station models to be solved. Morris also gives exact results for the case where priorities are reversed on both nodes.

The case of reversed priorities has an interesting application in the context of batch processing in operating systems. As an example, batch jobs in MVS are executed in initiator address spaces [Samso92]. Such an initiator is defined to process certain job classes in a well-defined priority order. A job class may as well be executed by multiple initiators. As such, it is perfectly possible that a job class receives highest priority at one initiator, while it has lowest priority at another. Closed queueing networks with one FCFS node (representing the pre- and postprocessing by the Job Entry Subsystem JES) and multiple priority nodes (representing the initiator address spaces) with reversed priorities may thus be well-suited to analyze the performance of such a batch processing system. Other applications may easily be found in the domain of operating systems (e.g. transaction processing in CICS), as well as in the domain of Client/Server and Distributed Processing (e.g. application replication at different nodes in a LAN in order to spread the load).

Strictly speaking, the model would have to be analyzed with non-preemptive servers and dynamic routing (a job is picked by a free initiator). However, as a starting point, this paper will concentrate on the analysis of a two-class three-node model with one FCFS server and two preemptive-resume priority servers with reversed priorities. We will further restrict the analysis - for simplicity only - to models with exponential class service times.

Although only few exact results are available for closed queueing networks with priorities, several approximations have been developed to analyze the performance of such networks. A brief overview of these approximations will be given in the next section. In the remainder of this paper, a clear definition of the model under investigation will be given, a possible solution algorithm will be discussed and the results will be compared to results of the application of another well-known approximation. Further solution approaches are discussed in less detail.

1. Approximation algorithms for closed queueing networks with priorities

Because of the complexity of these models and the lack of exact results, several approximations have been developed. Some of the approximations are based on the application of Norton's Theorem, a theorem that originates from electrical circuit theory. Chandy, Herzog and Woo [Chand75] prove that this theorem holds for, and gives exact results for, queueing networks that obey local balance. The analysis of such a network is greatly simplified by replacing part of it by a flow-equivalent service center. Using local balance solutions for networks that do not obey local balance may however introduce large inaccuracies. Sauer and Chandy [Sauer75] approximate the priority model by first coalescing the classes of the original model into three classes : a designated class and two composite classes, one of a higher and one of a lower priority than the designated class. Then Norton's Theorem is applied to solve the reduced model. This approach was proven to give satisfying results, very close to the exact results for an extensive number of tests.

Reiser [Reise76] uses exact techniques to solve a central server model with two priority classes at the (preemptive-resume) cpu. The approximation lies in the fact that the low priority class is served by a server whose capacity is reduced by the utilization attained by high priority jobs. Errors are introduced by the ignorance of the nonhomogeneity of the lower class processing times at the priority server and by the separate evaluation of both classes. Hierarchical decomposition (Norton's Theorem) is applied where necessary. As such, the priority network is approximated by a network not involving priorities and thus retaining the product form of the system state probabilities.

The idea of a separate server has been extended by Sevcik [Sevci77]. The Shadow Approximation provides in a 'shadow' server for the exclusive use of the low priority jobs. The service rate of the shadow server is slowed down to reflect the server utilization by high priority jobs. The model is evaluated as a multiple class model. High priority class utilization is determined by an efficient search method. The Shadow Approximation is therefore applicable to more general networks than those in [Reise76], e.g. in the case where different classes of jobs are allowed to have different priorities at different nodes.

Kaufman [Kaufm84] describes the errors that are induced by using the Shadow Approximation. One such error occurs because the shadow server is exclusively used for low priority jobs, allowing them to start processing immediately upon arrival at this server. In the original network however, low priority jobs will often have to wait for the completion of high priority jobs present in the queue at the moment the low priority job arrives there. This *delay error* is eliminated by application of Kaufman's Effective Service Approximation. The residual errors can partly be explained by the *variability* of the effective low priority service time, and by the assumption of the Arrival Theorem, which does not apply for the low priority job class at fcfs servers, resulting in the so-called *synchronization error*.

Other approximations have been developed by adjusting the MVA algorithm to include the effect of priorities [Bryan83][Bryan84]. These algorithms have the advantage of being computationally more efficient than non-MVA based approximations. Bryant et al. [Bryan83] develop an MVA approximation for preemptive and non-preemptive priority models. The response time formulas at the heart of MVA are replaced by response time formulas obtained from exact analysis of M/M/1 PR and HOL queues. The authors show that this approximation has an accuracy of within 5 % tolerance error for a large set of networks with one priority server. Eager and Lipscomb [Eager88] present an Approximate MVA Approximation, which is computationally more efficient than MVA and gives approximate results for priority networks that are acceptably close to the exact solution.

Bondi and Chuang [Bondi88] propose an MVA based approximation for a model with one preemptive server. The authors explicitly take into account that the Arrival Theorem is violated for the low priority class at the fcfs servers. A low priority job finishing service at the preemptive server, finds all higher priority jobs at the fcfs servers. These arrival instant queue lengths are calculated and used for the calculation of the low priority response time at the fcfs servers, as such reducing the synchronization error. Results are substantially better than for the Shadow Approximation. This shows that the accuracy is highly determined by the arrival instant queue lengths at both priority and fcfs servers, and by the accuracy with which the effective service times are predicted at the priority node.

None of the algorithms however, has been tested for models with multiple priority nodes and different priority structures at each node. The Shadow Approximation allows the analysis of such models, but with sometimes large errors. This paper presents a solution that is partly based on the Delay/MVA algorithm developed by Bondi and Chuang. In the next section, the model will be described and the algorithm explained. Section 3 compares the results with results that have been obtained by application of the Shadow Approximation. Our algorithm is clearly more accurate for the tested set of network parameters.

2. The model

Figure 1 represents the model under investigation. It consists of two preemptive resume (pr) servers and one fcfs server (the central server). Priorities are reversed on both pr servers, giving highest priority to class 1 jobs at server 2 and highest priority to class 2 jobs at server 3.

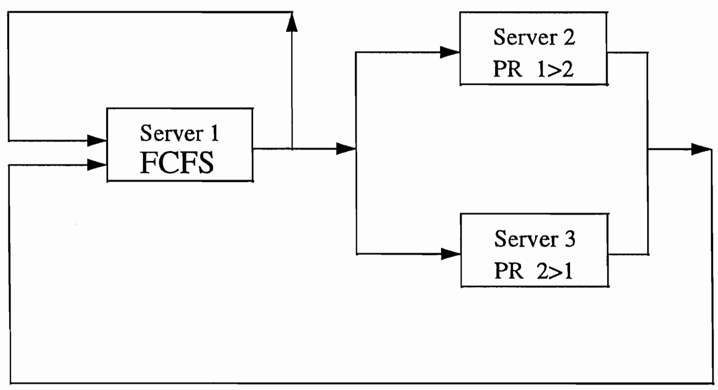

Figure 1 : basic central server priority model

Let both class service times on all servers be exponentially distributed with mean $s_{c,k}$ where c is the class index and k is the server index. It is assumed that the service times at the fcfs server (server 1) are class independent and equal to s_1. Let N_1 (N_2) be the class 1 (class 2) population and let $q_{c,k}(\vec{N})$ be the average number of class c jobs at server k assuming $\vec{N}=(N_1,N_2)$ jobs in the system. $q_{c,k}(\vec{N}-\vec{e}_i)$ then represents the average number of class c jobs at server k with one class i job removed from the system.

For the calculation of class response times at the priority servers, the exact M/M/1 priority queueing formulas are used. Since class 1 receives pr priority at server 2, and class 2 receives pr priority at server 3, we can write the response time formulas for class 1 at server 2 and class 2 at server 3 as follows :

$$R_{1,2} = s_{1,2}\left(1 + q_{1,2}(\vec{N} - \vec{e}_1)\right) \tag{2.1}$$

$$R_{2,3} = s_{2,3}\left(1 + q_{2,3}(\vec{N} - \vec{e}_2)\right) \tag{2.2}$$

Class 1 and class 2 response times at servers 3 and 2 respectively may then be written as

$$R_{2,2} = s_{2,2}\left(1 + q_{2,2}(\vec{N} - \vec{e}_2)\right)/\left(1 - \rho_{1,2}(\vec{N} - \vec{e}_2)\right) + s_{1,2}q_{1,2}(\vec{N} - \vec{e}_2) \tag{2.3}$$

$$R_{1,3} = s_{1,3}\left(1 + q_{1,3}(\vec{N} - \vec{e}_1)\right)/\left(1 - \rho_{2,3}(\vec{N} - \vec{e}_1)\right) + s_{2,3}q_{2,3}(\vec{N} - \vec{e}_1) \tag{2.4}$$

These formulas take into account the effective service time of the low-priority job at the pr server [Bondi88] and assume that the Arrival Theorem holds.

As mentioned before, the synchronization error occurs at the fcfs servers because all jobs of higher priority will be present there if a low-priority job arrives (violation of the Arrival Theorem). Therefore, the arrival instant queue length is not the equilibrium queue length.

The algorithm described in [Bondi88] therefore has been adapted to suit for the cs model with two preemptive servers. At the moment a class 2 job finishes service at server 2 and arrives at server 1, no class 1 job is present at server 2. All class 1 jobs will be present at server 1 and server 3. The class 1 queue lengths at these servers are then calculated using the Delay/MVA algorithm described by Bondi and Chuang. Service time and visit ratio of class 1 jobs at server 2 are set equal to 0, whereas the service time of the class 2 jobs is set equal to the effective service time so as to reflect the effect of preemption on class 2 service times. The resulting (reduced) model consists of one preemptive server and two fcfs servers (figure 2) and can now be solved using the Delay/MVA algorithm. As such, the arrival instant queue length of class 1 jobs at server 1 is calculated and used subsequently in the calculation of the response time of class 2 jobs at server 1.

Let $q^*_{1,1}$ be the number of class 1 jobs at server 1 upon arrival of a class 2 job, as calculated by Delay/MVA. Then the response time of class 2 jobs at server 1 will be

$$R_{2,1} = s_1\left(1 + q^*_{1,1} + q_{2,1}(\vec{N} - \vec{e}_2)\right) \tag{2.5}$$

Algorithm 1 :

```
- Initialize :    throughputs X_c
                  average queue lengths q_c,k
                  high priority device utilizations ρ_c,k
- For n_1 = 0 to N_1 do
    For n_2 = 0 to N_2 do
        * if both n_1 and n_2 > 0 then
                  - assume s_2,3 = 0 and v_2,3 = 0  (low pr job at server 3) and s_1,3 = s_1,3/(1-ρ_2,3);
                  calculate arrival instant queue length of class 2 jobs at server 1 using
                  Delay/MVA algorithm (Bondi)
                  - assume s_1,2= 0 (low pr job at server 3) and s_2,2 = s_2,2/(1-ρ_1,2);
                  calculate arrival instant queue length of class 2 jobs at server 1 using
                  Delay/MVA algorithm (Bondi)
        * calculate device response times for each class according to formulas
          described above (2.1-2.6)
        * calculate total response time and throughput for each class
        * calculate high priority device utilizations ρ_c,k and equilibrium queue lengths q_c,k
- End
```

3. The results

The algorithm has been tested for a number of network parameters as given in [Bondi88]. The results are compared with the results obtained from simulation of these models using GPSS/H. Simulation does not provide exact results; it is therefore recommended to derive confidence intervals for the point estimates. If the system is regenerative[1], such confidence intervals are estimated using the results of a number of independent 'tours' during one run. In our model, the order in which jobs are waiting at the fcfs server has an impact on the further evolution of the system. A regeneration point could possibly be defined by the order in which jobs are waiting at the fcfs server, but this cannot be derived from the simulation. Therefore, it has been opted to run the simulation for a number of replications (50), giving independent and identically distributed class response time results for which 95% confidence intervals have been calculated. Throughputs are derived from the average class response times by application of Little's Law.

Table 3.1 summarizes the data that are used for testing the algorithm. Table 3.2 shows the throughputs compared to the throughput point estimates from the simulations and to the results that have been obtained by application of the Shadow Approximation.

Algorithm 1 is for some models far more accurate than the Shadow Approximation : in 3 cases of 6, both class throughputs are within 5 % from the simulation results. Deviations of 10 % and more occur in 2 models for one class only and the maximum deviation is about 17 %. The Shadow Approximation shows only one model having both throughputs within 5 % of the simulation results. Deviations of 10 % and more

[1] A regenerative system contains regeneration points at which the system stochastically restarts. More about regenerative simulation may be found in [Crane74], [Laven75] and [Laven77].

occur in 4 of the 6 cases and the maximum is even about 44 %. Algorithm 1 therefore performs overall slightly better than the Shadow Approximation.

		Server 1		Server 2		Server 3	
		S	*V*	*S*	*V*	*S*	*V*
Model 1	class 1	0.01	21	0.036	10	0.036	10
	class 2	0.01	21	0.036	10	0.036	10
Model 2	class 1	0.01	13	0.036	8	0.020	4
	class 2	0.01	3	0.036	1	0.020	1
Model 3	class 1	0.01	3	0.036	1	0.020	1
	class 2	0.01	3	0.036	1	0.020	1
Model 4	class 1	0.10	6	0.360	1	0.360	4
	class 2	0.10	6	0.360	1	0.360	4
Model 5	class 1	0.01	15	0.036	10	0.036	4
	class 2	0.01	20	0.036	5	0.036	14
Model 6	class 1	0.01	13	0.036	8	0.036	4
	class 2	0.01	3	0.036	1	0.036	1

Table 3.1 : test data

		Simulation			Algorithm 1		Shadow	
		R		X				
Model 1	class 1	2.6162	± 0.0036	1.1467	1.0974	-0.043	1.2810	+0.117
	class 2	2.6164	±0.0030	1.1467	1.0974	-0.043	1.2810	+0.117
Model 2	class 1	0.9885	±0.0009	3.0348	3.1249	+0.030	3.1141	+0.026
	class 2	1.0362	±0.0016	2.8953	2.5702	-0.112	2.8558	-0.014
Model 3	class 1	0.1782	±0.0001	16.8363	16.3562	-0.029	17.3898	+0.033
	class 2	0.3822	±0.0005	7.8499	7.7169	-0.017	9.3062	+0.186
Model 4	class 1	52.7845	±0.3587	0.0568	0.0474	-0.165	0.0537	-0.055
	class 2	4.7424	±0.0079	0.6326	0.6457	+0.021	0.6407	-0.013
Model 5	class 1	2.3772	±0.0038	1.2620	1.3153	+0.042	1.1193	-0.113
	class 2	2.6470	±0.0038	1.1334	1.1276	-0.005	1.6364	+0.444
Model 6	class 1	1.1980	±0.0012	2.5042	2.5473	+0.017	2.7900	+0.114
	class 2	0.5441	±0.0008	5.5137	5.1657	-0.063	5.3165	-0.036

Table 3.2 : throughputs from simulation, algorithm 1 and shadow approximation

Algorithm 1 ignores the origin of the job arriving at the fcfs server. However, a class 1 job comes from server 2 with a probability of $v_{1,2}/(v_{1,2}+v_{1,3})$. The number of class 2 jobs it observes in this case is equal to the equilibrium queue length $q_{2,1}(\vec{N}-\vec{e}_1)$. If the same job finished at server 3 - which has a probability of $v_{1,3}/(v_{1,2}+v_{1,3})$ - then it would observe a class 2 queue length equal to $q^*_{2,1}$, as calculated by Delay/MVA. Therefore, the class 1 response time at server 1 could be written as

$$R_{1,1} = s_1\left[1 + q_{1,1}(\vec{N}-\vec{e}_1) + q^*_{2,1}v_{1,3}/\ (v_{1,2}+v_{1,3}) + q_{2,1}(\vec{N}-\vec{e}_1)v_{1,2}/(v_{1,2}+v_{1,3})\right] \tag{3.1}$$

Similarly, a class 2 job may leave server 2 with probability $v_{2,2}/(v_{2,2}+v_{2,3})$ and find a class 1 queue length at server 1 equal to $q^*_{1,1}$, as calculated by Delay/MVA, or leave server 3 with probability $v_{2,3}/(v_{2,2}+v_{2,3})$ and find a class 1 queue length at server 1 equal to the equilibrium queue length $q_{1,1}(\vec{N}-\vec{e}_2)$. Class 2 response time at server 1 could therefore be written as

$$R_{2,1} = s_1\left[1+q_{2,1}(\vec{N}-\vec{e}_2)+q^*_{1,1}v_{2,2}/(v_{2,2}+v_{2,3})+q_{1,1}(\vec{N}-\vec{e}_2)v_{2,3}/(v_{2,2}+v_{2,3})\right] \tag{3.2}$$

Replacing formulas (2.5) by (3.2) and (2.6) by (3.1) results in a slightly adapted algorithm (Algorithm 2), the results of which are given in table 3.3. In most of the cases, this algorithm is more accurate than Algorithm 1 and certainly more accurate than the Shadow Approximation. The maximum error is now reduced to 14.3 %, and this is the only case where the deviation from simulation exceeds 10 %.

		Simulation	Algorithm 2	
Model 1	class 1	1.1467	1.0981	-0.042
	class 2	1.1467	1.0981	-0.042
Model 2	class 1	3.0348	3.0907	+0.018
	class 2	2.8953	2.6849	-0.073
Model 3	class 1	16.8363	16.3923	-0.019
	class 2	7.8499	7.8368	-0.026
Model 4	class 1	0.0568	0.0487	-0.143
	class 2	0.6326	0.6438	+0.018
Model 5	class 1	1.2620	1.1568	-0.083
	class 2	1.1334	1.1309	-0.002
Model 6	class 1	2.5042	2.5195	+0.006
	class 2	5.5137	5.2715	-0.044

Table 3.3 : throughputs obtained from simulation and algorithm 2

4. Arrival Theorem Assumption

The main contribution of the Delay/MVA algorithm and our algorithms 1 and 2, is the explicit calculation of instant arrival queue lengths at the fcfs server(s) in order to reduce the synchronization error. This error occurs if the network is analyzed using Shadow Approximation, because it assumes that the Arrival Theorem applies for the low priority class although it does not. The Arrival Theorem [Sevci81] states that a job arriving at a server sees the equilibrium distribution of the network states if that job belongs to an open class. A job belonging to a closed class sees the equilibrium distribution of the network with one job of that class removed.

It is clear that this Theorem does not hold for low priority class jobs in a priority network for the reason explained above (section 2). However, since in our model priorities are reversed on both priority servers, it might well be that the Arrival Theorem holds 'nearly' : the effect of receiving highest priority on one server is compensated by the effect of receiving lowest priority on the other server.

If this is true, using the equilibrium queue lengths for class response time calculations at the fcfs server would result in reasonable approximations for the performance measures. The response time formulas at the fcfs server then reduce to the classical MVA expressions

$$R_{1,1} = s_1\left(1 + q_{1,1}(\vec{N} - \vec{e}_1) + q_{2,1}(\vec{N} - \vec{e}_1)\right) \tag{4.1}$$

$$R_{2,1} = s_1\left(1 + q_{1,1}(\vec{N} - \vec{e}_2) + q_{2,1}(\vec{N} - \vec{e}_2)\right) \tag{4.2}$$

The algorithm therefore reduces to a simple MVA algorithm where class response times at the priority servers are calculated according to the priority formulas (2.1-2.4) and where class response times at the fcfs server are simply calculated using the normal MVA expressions (4.1-4.2) (showing obviously much similarity with The MVA Approximation [Bryan83] and [Bryan84], but extended for the case of multiple priority centers). The resulting algorithm is called here Algorithm 3. The results are summarized in table 4.1.

		Simulation	Algorithm 3	
Model 1	class 1	1.1467	1.0993	-0.041
	class 2	1.1467	1.0993	-0.041
Model 2	class 1	3.0348	3.1041	+0.023
	class 2	2.8953	2.5949	-0.104
Model 3	class 1	16.8363	16.6934	-0.008
	class 2	7.8499	7.8557	+0.001
Model 4	class 1	0.0568	0.0462	-0.187
	class 2	0.6326	0.6471	+0.023
Model 5	class 1	1.2620	1.1609	-0.080
	class 2	1.1334	1.1314	-0.002
Model 6	class 1	2.5042	2.5170	+0.005
	class 2	5.5137	5.3001	-0.039

Table 4.1 : throughputs obtained from simulation and algorithm 3

The results do not differ significantly from the results obtained from Algorithm 2. This may lead us to a first simple conclusion that the synchronization error is not significant in this type of model, and that the Arrival Theorem 'nearly' holds for the low priority class(es). In that case, Algorithm 3 is much more performant than the other algorithms, since low class arrival instant queue lengths at the fcfs server do not have to be explicitly calculated.

5. Example models with non-homogeneous service times

The test models as evaluated in section 3 are restrictive in the sense that the service times at the pr servers are not class dependent, but server dependent. An additional set of models with non-homogeneous service times has been evaluated. The data are given in table 5.1, results are summarized in table 5.2 and 5.3.

			Server 1		Server 2		Server 3	
		N	*S*	*V*	*S*	*V*	*S*	*V*
model 7	class 1	5	0.01	9	0.02	4	0.09	4
	class 2	5	0.01	3	0.15	1	0.04	1
model 8	class 1	6	0.01	7	0.01	4	0.02	2
	class 2	3	0.01	11	0.20	2	0.012	8
model 9	class 1	10	0.01	6	0.005	4	0.01	1
	class 2	10	0.01	6	0.01	1	0.005	4
model 10	class 1	8	0.005	3	0.01	1	0.03	1
	class 2	5	0.005	11	0.03	7	0.01	3
model 11	class 1	10	0.1	3	0.1	1	0.2	1
	class 2	10	0.1	3	0.2	1	0.1	1

Table 5.1 : additional test data

		Simulation			Algorithm 1		Shadow	
		R		X				
Model 7	class 1	2.3173	± 0.0031	2.1577	2.1145	-0.020	2.1632	+0.003
	class 2	0.9093	±0.0010	5.4989	5.4963	-0.001	5.5040	+0.001
Model 8	class 1	0.5039	±0.0002	11.9072	11.4648	-0.037	11.9398	+0.003
	class 2	2.5234	±0.0038	1.1889	1.1873	-0.001	1.2162	+0.023
Model 9	class 1	1.2015	±0.0005	8.3231	8.2434	-0.010	8.3333	+0.001
	class 2	1.2017	±0.0004	8.3214	8.2434	-0.009	8.3333	+0.001
Model 10	class 1	0.2685	±0.0001	29.7994	29.7510	-0.002	29.8296	+0.001
	class 2	1.5050	±0.0013	3.3222	3.2584	-0.019	3.3310	+0.003
Model 11	class 1	6.0310	±0.0387	1.6581	1.6448	-0.008	1.6667	+0.005
	class 2	6.0046	±0.0041	1.6654	1.6448	-0.012	1.6667	+0.001

Table 5.2 : throughputs from simulation, algorithm 1 and shadow approximation

		Simulation	Algorithm 2		Algorithm 3	
model 7	class 1	2.1577	2.1144	-0.020	2.1144	-0.020
	class 2	5.4989	5.4966	-0.001	5.4967	-0.001
model 8	class 1	11.9072	11.7061	-0.017	11.8955	-0.001
	class 2	1.1889	1.1877	-0.001	1.1819	-0.006
model 9	class 1	8.3231	8.3149	-0.001	8.3333	+0.001
	class 2	8.3214	8.3149	-0.001	8.3333	+0.001
model 10	class 1	29.7994	29.7489	-0.002	29.7434	-0.002
	class 2	3.3222	3.2581	-0.019	3.2577	-0.019
model 11	class 1	1.6581	1.6551	-0.002	1.6666	+0.005
	class 2	1.6654	1.6551	-0.006	1.6666	+0.001

Table 5.3 : throughputs from simulation, algorithm 2 and algorithm 3

It is remarkable that, although the results of all 3 algorithms are very close to the simulation results, the Shadow Approximation totally outperforms the other 3 algorithms for these data. In all models, the high priority service time is small compared to the low priority service time. Morris [Morri81] stated that this is a sufficient condition to be satisfied if the Shadow Approximation is to be a good approximation (within 2% errors), for the delay error is small in that case. This statement now seems also to be true in the type of model considered here. It is furthermore remarkable how both Algorithm 1 and 2 consistently (slightly) underestimate the throughput, while the Shadow Approximation overestimates it. Again, Algorithm 3 provides a faster way to obtain approximately the same accuracy as Algorithms 1 and 2 for this type of model.

6. Conclusions and future research

The research in this paper suggests that the case of closed queueing networks with multiple preemptive (or non-preemptive) nodes and different priority structures at each node presents a model with interesting applications in practice, which has received little attention in literature. The algorithm developed here solves the three-node two-class preemptive model with reasonable accuracy, giving better results than the Shadow Approximation in some cases. The typical structure of this model however also suggests that the Arrival Theorem 'nearly' holds, which simplifies the solution procedure to a classical MVA solution, almost similar to The MVA Approximation proposed in [Bryan83] and [Bryan84]. For models with more than two priority servers and more than two priority classes, it will have to be determined to what degree the Arrival Theorem does or does not hold. This will largely depend on the priority structures at each of the priority nodes. Algorithm 2 will be the most general solution to such models in which the Arrival Theorem doesn't even hold nearly. Algorithm 3 provides a faster solution for these models with approximately the same accuracy.

The next step in this research is to make the algorithm applicable for more general models with more than two priority nodes and more than two priority classes. The final objective is to provide a framework for determining optimal priority structures in batch and client/server environments. As such, the contribution of this paper may be viewed as a step towards the development of such a framework.

References

[Avi-I73] Avi-Itzhak B. And D.P. Heyman, 'Approximate Queueing Models for Multiprogramming Computer Systems', *Operations Research*, Vol. 21, No. 6, 1973, pp. 1212-1230.

[Baske75] Baskett F., K.M. Chandy, R.R. Muntz and F.G. Palacios, 'Open, Closed and Mixed Networks of Queues with Different Classes of Customers', *Journal of the ACM*, Vol. 22, No. 2, April 1975, pp. 248-260.

[Bondi88] Bondi A.B. and Y-M. Chuang, 'A New MVA-Based Approximation for Closed Queueing Networks with a Preemptive Priority Server', *Performance Evaluation*, Vol. 8, 1988, pp. 195-221.

[Bryan83] Bryant R.M., A.E. Krzesinski and P. Teunissen, 'The MVA pre-empt resume priority approximation', *Proceedings of the 1983 ACM Sigmetrics Conference*, 1983, pp. 12-27.

[Bryan84] Bryant R.M., A.E. Krzesinski, M.S. Lakshmi and K.M. Chandy, 'The MVA Priority Approximation', *ACM Transactions on Computer Systems*, Vol. 2, No. 4, November 1984, pp. 335-359.

[Buzen73] Buzen J.P., 'Computational algorithms for closed queueing networks with exponential servers', *Communications of the ACM*, Vol. 16, No. 9, September 1973, pp. 527-531.

[Chand75] Chandy K.M., U. Herzog and L. Woo, 'Parametric Analysis of Queueing Networks', *IBM Journal of Research and Development*, January 1975, pp. 36-42.

[Crane74] Crane M.A. and D.L. Iglehart, 'Simulating Stable Stochastic Systems, II : Markov Chains', *Journal of the ACM*, Vol. 21, No. 1, January 1974, pp. 114-123.

[Eager88] Eager D.L. and J.N. Lipscomb, 'The AMVA Priority Approximation', *Performance Evaluation*, Vol. 8, 1988, pp. 173-193.

[Kaufm84] Kaufman J.S., 'Approximation Methods for Networks of Queues with Priorities', *Performance Evaluation*, Vol. 4, 1984, pp.183-198.

[Laven75] Lavenberg S.S. and D.R. Slutz, 'Introduction to Regenerative Simulation', *IBM Journal of Research and Development*, September 1975, pp. 458-462.

[Laven77] Lavenberg S.S. and C.H. Sauer, 'Sequential Stopping Rules for the Regenerative Method of Simulation', *IBM Journal of Research and Development*, November 1977, pp. 545-558.

[Morri81] Morris R.J.T., 'Priority Queueing Networks', *The Bell System Technical Journal*, Vol. 60, No. 8, October 1981, pp. 1745-1769.

[Reise76] Reiser M., 'Interactive modeling of computer systems', *IBM Systems Journal*, No. 4, 1976, pp. 309-327.

[Reise80] Reiser M. and S.S. Lavenberg, 'Mean Value Analysis of closed multichain queueing networks', *Journal of the ACM*, Vol. 27, No. 2, April 1980, pp. 313-322.

[Samso92] Samson S.L, *MVS Performance Management (ES/9000 edition)*, McGraw-Hill, New York, 1992, 365 pp.

[Sauer75] Sauer D.H. and K.M. Chandy, 'Approximate Analysis of Central Server Models', *IBM Journal of Research and Development*, May 1975, pp. 301-313.

[Sevci77] Sevcik K.C., 'Priority Scheduling Disciplines in Queueing Network Models of Computer Systems', *IFIP Information Processing 77*, 1977, pp. 565-570.

[Sevci81] Sevcik K.C. and I. Mitrani, 'The distribution of Queueing Network States at Input and Output Instants', *Journal of the ACM*, Vol. 28, No. 2, April 1981, pp. 358-371.