The design of a system with two parallel queues and one-way overflow

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computing Science

Memorandum COSOR 83 - 02

The design of a system with two parallel queues and one-way overflow

by

J.B.M. van Doremalen

Eindhoven, the Netherlands

January 1982

THE DESIGN OF A SYSTEM WITH TWO PARALLEL QUEUES AND ONE-WAY OVERFLOW

by

J.B.M. van Doremalen

Abstract

This paper deals with the problem how to design a queueing system, where two independent Poisson arrival streams have to be served efficaciously. The system consists of two parallel multiserver queues with waiting-rooms in which one-way overlow is allowed for.

First a concise analysis for both exact and approximative methods in the system with overlow is given. Then we show that in a specific situation a conversational computer algorithm might be an efficient instrument in the design of such systems. Special attention is devoted to a particular step in the algorithm: the solution of an optimization problem. An enumeration algorithm, using the proposed approximations, is discussed.
Eventually some conclusions are drawn.

1. Introduction

This paper deals with the problem how to design a queueing system, where two types of customers arrive according to independent Poisson processes with parameters λ_1 and λ_2 respectively. It is assumed that both types of customers ask for the same negative expontentially distributed service time with mean μ^{-1}. However, there are type dependent requirements, e.q. with respect to loss probabilities and average waiting times.

Two opposing aims arise. On one hand the installation and maintenance costs for the system have to be minimized, but on the other hand the arriving customers have to be served efficaciously.

Our interest is in systems, which consist of two parallel multiserver queues with finite waiting rooms and in which the installation of a one-way overflow mechanism has to be considered. The importance of such systems is paramount as overflow mechanisms are being installed more and more, for instance in the field of telecommunication networks (confer Kuczura [6], Rath and Sheng [8] and Morrison [7]).

We will analyse explicitly a design problem where maintenance costs, depending on the configuration of the system, play a role along with constraints for the utility factors and two customer dependent quantities, the loss probabilities and average waiting times. As an example of such a system we mention the reservation system of an airline company, where travel-agencies and individual customers are being served by two groups of operators. Calling travel agencies are allowed to overflow to the second group of operators under congested conditions for the first group.

First we will analyse the system with two parallel queues and one-way overflow. In Section 2 both exact and approximative methods to evaluate steady-state quantities in such systems are discussed.

Then the design problem is studied. We state that a conversational approach has to be recommended in order to find satisfying queueing systems. This observation is based on the following arguments. A designer, who is looking after an efficient system, in general will not be able to formulate his wishes in precise detail. Another problem is that it is very difficult, if not impossible, to foresee what the consequences are of the conflicting aims the designer sets. In Section 3 we will return to this subject and sketch the framwork for a conversational approach in our specific problem.

Our main concern is with a particular step in the approach, namely the formulation and solution of an optimization problem. In section 4 an enumeration algorithm to solve this problem is described. The efficient use of approximations for steady-state quantities becomes apparent.

Eventually, in Section 5, some concluding remarks are made.

2. The evaluation of steady-state quantities in a queueing system with one-way overflow

In this section a queueing system, where overflow is allowed for, is analysed. The system consists of two finite queues with s_ℓ servers and n_ℓ waiting places, $\ell = 1,2$. At each queue customers arrive according to independent Poisson processes with rates $\lambda_\ell > 0$, $\ell = 1,2$. The service times are independent and negative exponentially distributed with the same mean μ^{-1} for all customers.

Customers arriving at the first queue are allowed for to overflow to the second queue, if all service units and waiting places are occupied. In both queues the service discipline is first-come first-served irrespective of the type of customer, i.e. there is no priority for overflowing customers at the second queue.

The system can be described as a continuous-time Markov process on a finite two-dimensional state space. The steady-state probabilities $p_{i,j}$, that i customers are in the first queue and j in the second, are determined uniquely by a set of two-dimensional birth-and-death equations and a normalization. The steady-state quantities, loss probabilities and average waiting times for both types of customers and service utility factors in both queues, are simple functions of the $p_{i,j}$'s.

In a preceding paper [3] we have discussed a block iterative method to obtain the steady-state probabilities, based on the algebraic structure of the transition matrix. Other methods based on a numerical solution of the equilibrium equations are discussed in Morrison [7] and Brandwajn [1]. Approximations for the steady-state quantities, using this approach, have to be based on decomposition ideas for the transition matrix.

A more fertile approach to obtain approximations is based on a separation argument. The first queue can be analysed independently as a M/M/s/k system (a queue with Poissonian input, s exponential servers and k - s waiting places). The overflow process of the queue is a renewal process and can be shown to have a hyperexponentially distributed inter-overflow-time (confer van Doorn [2]).

The second queue then can be analysed as a system with two independent arrival streams, a Poissonian main stream of type 2 customers and a renewal stream of

overflowing type 1 customers. This analysis has been carried out in Kuczura [6] and Rath and Sheng [8] have shown that the Interrupted Poisson process (IPP), with aptly chosen parameters, is a good approximation for the overflow process of a M/M/s/k queue. The IPP is a Poisson process which is alternately turned on for an exponentially distributed time and then turned off for another (independent) exponentially distributed time. The parameters are evaluated matching the first two moments of the overflow process and the IPP.

Eventually, we observe that a first order approximation (the IPP can be seen as a second order approximation) can be obtained by identifying the overflow process with a Poisson process.

Numerical experiments have shown that the first order approximation is rather accurate for the evaluation of the average waiting times for both types of customers, the loss probability for type 2 customers and the service utility factors for both queues. The second order approximation is rather accurate for the evaluation of the loss probability of type 1 customers.
In Section 4 we will see that these approximations can be used efficiently in the solution of an optimization problem for the queueing system with one-way overflow.

3. The design algorithm

In the previous section we have seen how queueing systems with one-way overflow can be analysed. Now we will discuss the design problem. As has been pointed out in the introduction a conversational approach has to be recommended for several reasons. In this section the idea will be worked out in more detail.

Generally, the designer - looking for an efficient queueing system - will not be able to formulate his wishes in precise detail, as he has only a rude idea of the real problems and possibilities of the system. It therefore is not possible to solve the design problem in a one-step optimization procedure, as there will be no unique optimization problem reflecting all the explicit and implicit wishes of the designer. One of the reasons for this difficulty is that there are conflicting aims. It will not be clear on forehand what the consequences will be of setting certain goals. Another problem is that certain requirements, for instance with respect to the sensitivity of the system, are very hard to represent in terms of an optimization problem.

A reasonable approach to the problem might be as follows. The designer makes an analysis of the costs and requirements of the system and formulates a first draft for an optimization problem. This problem has to be solved.

Then the designer evaluates the optimal solution of the problem. He makes a sensitivity analysis and evaluates the consequences of his earlier analysis, for example with respect to the goals he has set himself. If need be he reformulates the optimization problem according to his new insights and tries to obtain new information in a subsequent step of his design procedure.

Concluding we might say that for finding a satisfactory design there has to be the possibility of a constant reformulation, evaluation and recapitulation of the gathered information so far. A conversational approach, in whatever form, therefore seems the obvious means to solve the design problem satisfactory, as the designer has to play an important controlling role in the designing process.

In the remaining part of this paper we will go into an important step of the design procedure: the optimization problem. We will restrict ourselves to the designing problem mentioned in the introduction. An important consequence of the conversational approach is that on the one hand the optimization problem need not be very refined and its solution not necessarily optimal, but that on the other hand the problem must have a rather flexible structure in order to be able to reflect a wide scale of possible aims. We will shown such an optimization problem in our situation.

Let us suppose that maintenance costs, depending on the configuration of the system and denoted as $f(n_1,s_1,n_2,s_2)$, play a role along with constraints for the service utility factors, denoted as $C_j(n_1,s_1,n_2,s_2)$ for queue $j = 1,2$, and for the loss probabilities and average waiting times of accepted customers, denoted as $L_i(n_1,s_1,n_2,s_2)$ and $W_i(n_1,s_1,n_2,s_2)$ for type i customers, $i = 1,2$.

An optimization problem reflecting these ideas for costs and requirements might be the following for given instream rates λ_1 and λ_2 and service rate μ,

P1: minimize $f(n_1,s_1,n_2,s_2)$

subject to:

1. $0 \leq n_j \leq N_j$, $1 \leq s_j \leq S_j$, $j = 1,2.$
2. $L_i(n_1,s_1,n_2,s_2) < \varepsilon_i$, $i = 1,2.$
3. $W_i(n_1,s_1,n_2,s_2) < \delta_i$, $i = 1,2.$
4. $c_j(n_1,s_1,n_2,s_2) < \gamma_j$, $j = 1,2.$

In the next section we will describe an enumeration algorithm to solve for this problem. We note that the problem is determined by the cost-function f, the physical limitations N_j and S_j, $j = 1,2$, and the constraints for loss probabilities, average waiting times and service utility factors ε_i, δ_i, $i = 1,2$ and γ_j, $j = 1,2$.

4. An enumeration algorithm

This section is devoted to an enumeration algorithm to solve the optimization problem P1. It should be noted that the problem is complicated by the fact that constraints are difficult to evaluate and the structure of the feasible region is not clear. As has been pointed out by Kovács [4:33], an enumeration becomes a method only, if a high proportion of solutions is implicitly examined. Here an efficient use of approximations has to be made also, as the inspection of the feasibility of a single solution, using an exact method, consumer relatively much computing time. The enumeration algorithm to be described is based on the separation argument, which in Section 2 was said to lead to good approximations.

As has been pointed out, the first queue can be analysed as a M/M/s/k system. Given a configuration (n_1, s_1) for the first queue, the second queue can be analyzed using the approximation ideas of Section 2. We therefore propose an enumeration of the possible configurations of the first queue and embed an enumeration for the second queue.

Before we give a more detailed description of the enumeration algorithm, some assumptions for the objective function f are discussed. Furthermore, a problem related to P1 for the M/M/s/k system is solved, as it will turn out to be a useful tool in the enumeration algorithm.

We will assume that the objective function f satisfies the following separation and monotonicity requirements

A1. $f(n_1,s_1,n_2,s_2) = f_1(n_1,s_1) + f_2(n_1,s_1,n_2,s_2)$

A2. $f_1(n_1,s_1) < f_1(n_1 + 1,s_1)$

A3. $f_1(n_1,s_1) < f_1(n_1,s_1 + 1)$

A4. $f_2(n_1,s_1,n_2,s_2) < f_2(n_1,s_1n_2 + 1,s_2)$

A5. $f_2(n_1,s_1,n_2,s_2) < f_2(n_1,s_1,n_1,s_2 + 1)$.

The assumptions are, though not unrealistic, not necessarily satisfied. For instance, if costs for losing customers of type 1 are involved, A4 need not be true. It will be clear that the assumptions are needed in order to obtain an efficient enumeration of the possible configurations based on the separation argument. The first queue now can be analyzed with the objective function f_1, and the second queue can be analyzed with the objective function f_2, if a fixed configuration (n_1,s_1) of the first queue is given.

An important subproblem in the enumeration algorithm is an optimization problem for a M/M/s/k system related to problem P1. Let $L(n,s)$, $W(n,s)$ and $C(n,s)$ denote the loss probability, the average waiting time for accepted customers and the service utility factor of a M/M/s/k system with s servers and n waiting places. And let $f(n,s)$ denote the objective function satisfying assumptions A2 and A3. The subproblem P2 then is, for given instream rate λ and service rate μ:

P2: minimize $f(n,s)$

under 1. $0 \leq n \leq N \quad 1 \leq s \leq S$

2. $L(n,s) < \varepsilon \quad W(n,s) < \delta \quad C(n,s) < \gamma$.

A simple enumeration algorithm to solve problem P2 is based on the monotonicity assumptions for f and the following observations

(1) $L(n,s) > L(n+1,s)$

(2) $W(n,s) < W(n+1,s) \quad W(0,s) = 0$

(3) $C(n,s) < C(n+1,s)$

(4) $C(0,s) > C(0,s+1)$

(5) $\lim_{n\to\infty} L(n,s) = \max\left\{0, 1 - \frac{\mu s}{\lambda}\right\}$

and can be given in quasi-algol as

```
begin
    s := 1;  fmin := +∞ ;
    while s ≤ S and C(0,s) ≥ γ and 1 - μs/λ ≥ ε do s := s + 1;
    while f(0,s) < fmin and s ≤ S do
    begin
        n := 0;
        while f(n,s) < fmin and L(n,s) ≥ ε and W(n,s) < δ and C(n,s) < γ
              and n ≤ N
        do n := n + 1;
        if f(n,s) < fmin and L(n,s) < ε and W(n,s) < ε and C(n,s) < γ
              and n ≤ N
        then begin fmin := f(n,s);  nmin := n;  smin := s end;
        s := s + 1
    end
end.
```

At the end of the procedure the optimal configuration is (nmin, smin) with corresponding objective value fmin. If fmin = +∞, then there is no feasible configuration.

Now we can start with an enumeration algorithm for problem P2. First, the enumeration of the configurations for queue 1 is discussed. We propose to use a natural ordering in the enumeration. The property which most prominently influences the second queue is the intensity of the overflow process. We therefore consider an enumeration of configurations with a decreasing overflow intensity and, consequently, an increasing objective value for f_1. Note that, unfortunately, the overlow process is not determined by the intensity only.

If ε_{min} and ε_{max} denote the minimal, respectively maximal overflow intensity we want to consider, the enumeration of P2 becomes

```
begin
    ε1^(0) := εmax; ν := 0; f min := +∞
    while ε1^(ν) ≥ εmin do
    begin
        solve the following P2 problem with instream rate λ1 and service rate μ1
        minimize f1(n1,s1)
        under 1. 0 ≤ n1 ≤ N1 and 1 ≤ s1 ≤ S1
              2. L(n1,s1) < ε1^(ν)  W(n1,s1) < δ1*  C(n1,s1) < γ1
        with optimal point (n1^(ν),s1^(ν));
        start the embedded enumeration of the second system, the optimal
        point is (n1^(ν),s1^(ν),n2^(ν),s2^(ν)) value f^(ν);
        if f^(ν) < f min
        then begin f min := f^(ν);   min := (n1^(ν),s1^(ν),n2^(ν),s2^(ν)) end;
        ε1^(ν+1) := L(n1^(ν),s1^(ν)); ν := ν + 1
    end
end.
```

We note that the P2 problem is rather easy to solve, as we can use the configuration $(n_1^{(\nu)}, s_1^{(\nu)})$ as a starting point of the enumeration in step $\nu + 1$.

An open question is the choice of the parameter δ_1^*. It should be noted that the average waiting time for accepted customers of type 1 can be split in two parts: customers accepted at queue 1 and queue 2 respectively. If $\delta_2 >> \delta_1$, then overflow is to be dissuaded. In other cases $\delta_1^* = \delta_1$ should have to be a reasonable choice.

Eventually the embedded enumeration has to be discussed. The problem to be solved is given by

P3: minimize $f_2(n_1^{(\nu)}, s_1^{(\nu)}, n_2, s_2)$

subject to:

1. $0 \leq n_2 \leq N_2 \quad 1 \leq s_2 \leq S_2$.
2. $L_i(n_1^{(\nu)}, s_1^{(\nu)}, n_2, s_2) > \varepsilon_i \quad , i = 1,2.$
3. $W_i(n_1^{(\nu)}, s_1^{(\nu)}, n_2, s_2) < \delta_i \quad , i = 1,2.$
4. $C_2(n_1^{(\nu)}, s_1^{(\nu)}, n_2, s_2) < \gamma_2$

where the instream rates λ_i, $i = 1,2$ and the service rate μ are given.

In Section 2 it is noted that L_2, W_2 and C_2 can be approximated rather good assuming that the second queue behaves as a M/M/s/k system with an adjusted instream rate $\lambda^{(\nu)} = \lambda_2 + \lambda_1 L(n_1^{(\nu)}, s_1^{(\nu)})$ (the first order approximation). This brings the formulated problem P3 back to a P2-problem, where, in case a feasible - possibly better - configuration has been found, the remaining restrictions $W_1 < \delta_1$ and $L_1 < \varepsilon_1$ have to be verified. This can be done using the more refined second order approximation or an exact method.

It should be noted that the following monotonicity arguments can be used,

$$L_1(n_1,s_1,n_2,s_2) > L_1(n_1,s_1,n_2+1,s_2)$$
$$W_1(n_1^{(\nu)},s_1^{(\nu)},0,s_2) < W(n_1^{(\nu)},s_1^{(\nu)}) < \delta_1^*$$
$$W_1(n_1,s_1,n_2,s_2) > W_1(n_1,s_1,n_2-1,s_2) \Rightarrow W_1(n_1,s_1,n_2+1,s_2) > W_1(n_1,s_1,n_2,s_2).$$

We now have an anumeration algorithm to solve the original optimization problem. As a result we have that $f_1^{(\nu)}$ is monotonously increasing and that $f_2^{(\nu)}$ is monotonously decreasing. As both the enumeration of (n_1,s_1) and the embedded enumeration are finite, the optimal solution is found in a finite number of steps.

It is, of course, possible to accelerate the scheme substantial. But it is not our purpose to analyse all kinds of refinements for the algorithm. What we wanted to show is the efficient use of separation arguments in designing such enumeration algorithms. Once more, however, we emphasize the fact that the found solution is not necessarily the optimal point of the original P2 problem, due to the kind of enumeration of the first queue's configurations and due to the use of approximations for the steady state quantities.

5. Conclusions

The purpose of this paper was to show how queueing systems with one-way overflow could be desgined. We think, that by analysing a specific problem, we have given a good insight in the nature of the problems that one encounters in such an analysis. Furthermore, the techniques, we have demonstrated, seem to be much more generally applicable.

Numerical experiments showed that the discussed problem could be handled rather efficiently for relatively large problems, say for instance 50 servers and waiting places in each queue. Solving exactly for the steady-state quantities in systems with that size is very time consuming, and one has to rely on the approximations completely. If one wants a very detailed and precise analysis this is a drawback. Otherwise the methods described in this paper seem very useful.

6. References

[1] Brandwajn, A. (1979): "An iterative solution of two-dimensional birth-and-death processes", Oper. Research, vol. 27: 595-605.

[2] van Doorn, E. (1982): "A note on the overflow process from a finite Markovian queue", private communications.

[3] van Doremalen, J. and J. Wessels (1982): "On two parallel queues with one-way overflow", Memorandum COSOR 82-15, University of Technology Eindhoven.

[4] Kovács, L. (1980): "Combinatorial methods of discrete programming", Akadémiai Kiadó, Budapest.

[5] Kaczura, A. (1972): "Queues with mixed renewal and Poisson inputs", The Bell System Techn. J., vol. 51: 1305-1326.

[6] ---------- (1973): "The interupted Poisson process as an overflow process", The Bell System Techn. J., vol. 52: 437-448.

[7] Morrison, J. (1981): "An overflow system in which queueing takes precedence", The Bell System Techn. J., vol. 60: 1-12.

[8] Rath J. and D. Sheng (1979): "Approximations for the overflow from queues with a finite waiting room", Oper. Research, vol. 27: 1208-1216.