Theoretical Computer Science 125 (1994) 111–130
Elsevier

Exact solutions for networks of queues with blocking-after-service*

Ian F. Akyildiz and Horst von Brand**

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Abstract

Akyildiz, I.F. and H. von Brand, Exact solutions for networks of queues with blocking-after-service, Theoretical Computer Science 125 (1994) 111–130.

The paper has two major parts. The first part deals with two-station networks with blocking-after-service (BAS) mechanism and different station types. In this part only single class of jobs is allowed. The contribution here is to show that exact solutions exist for two-station queueing networks with BAS mechanism having different station types. The exact equilibrium state probability distributions are derived. Insensitivity is investigated and formulas for performance measures are obtained. It is demonstrated that the throughput, mean number of jobs and mean number of Cblocked jobs depend on the scheduling discipline. A queueing network model with more than two stations is analyzed in the second part. Multiple-job classes with job class change, and different station types are allowed in the model. Exact solutions for equilibrium state probabilities and performance measures are obtained under the condition that only a certain total number of jobs is allowed in the network.

1. Introduction

Queueing networks have been used increasingly as tools for performance evaluation of computer systems, manufacturing systems and communication networks. For some special cases, which we call *classical networks*, the exact probability distributions are known [9, 15] and efficient algorithms [13] can be used to compute performance measures such as throughput and mean number of jobs.

Correspondence to: I.F. Akyildiz, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA.

* This work was supported by National Science Foundation (NSF) under Grant No. CCR-90-11981.

** Permanent address: Departamento de Informática, Universidad Técnica Federico Santa María, Casilla 110-V, Valparaíso, Chile.

SSDI 0304-3975(93)E0118-N

Classical networks assume that the capacity of the stations offering service is infinite, an assumption that usually does not hold in the actual systems. This gives rise to *queueing networks with blocking*. In recent years there has been an increased interest in the analysis of queueing networks with blocking. This is probably due to the realization that these queueing networks are useful in modeling computer systems, communication networks, and flexible manufacturing systems. A special issue [4] appeared in a journal which gives the state-of-the-art in this research area. The set of rules that dictate when a node becomes blocked and unblocked is commonly referred to as the *blocking* mechanism. There are basically only a few blocking mechanisms that have been extensively studied in the literature [4]. The blocking type we investigate in this work is called type-1 blocking, transfer blocking, production blocking and nonimmediate blocking. The naming of blocking mechanisms has been standardized in [4], where this case received the name *blocking-after-service* (BAS) where a job upon service completion at station i attempts to join destination station j. If station j at that moment is full, the job is forced to wait in the server of station i, until it enters destination station j. The server remains blocked for this period of time and it cannot serve any other job waiting in the station.

There are very few exact results for systems with BAS. A two-station model was considered by Akyildiz [1] for first-come-first served (FCFS) service only. Akyildiz shows that there is a nonblocking queueing network with the same transition matrix as the blocking network after relabeling of states, so the model is solved exactly. Balsamo and Donatiello [8] showed exact solutions for cycle time distributions for two-node closed queueing networks with blocking. The basic ideas are taken from [1], a nonblocking queueing network can be found with appropriate total number of jobs which has a product form solution. Onvural and Perros [20] consider closed queueing networks with more than two stations and BAS for limited number of jobs. Onvural [17] discusses product form solutions of several models with BAS. A recent survey of the area of queueing networks with blocking is given by Onvural [18].

Our literature study reveals that existing or proposed methods either contain disadvantages (e.g., long-run times or memory space) or restrictions (only two-node or tandem network solutions) or provide approximate results which differ widely from the exact values. In case of closed queueing networks the queueing network models with BAS mechanism have the additional limitations such as that all service time distributions are exponential, the queueing discipline at each node is basically FCFS and all jobs belong to the same class. Most of the models of the existing systems show these characteristics. Therefore, in this paper we investigate these limitations and obtain exact solutions for certain queueing networks with BAS mechanism. The paper has essentially two major parts. The first part deals with two-station networks with BAS mechanism and different station types. In this part we allow only a single class of jobs. The contribution here is to show exact solutions do exist for two-station queueing networks with BAS mechanism having different station types. We were not able to extend these results to multiple job class case. In the second part we allow multiple job classes and different station types.

We also permit more than two stations in the network. However, we introduce a condition on the number of permitted jobs in the network so that we can get exact solutions for the model. The paper is organized as follows. In Section 2 we analyze the two-station model. First we describe the model in detail. Then we obtain the exact equilibrium state probability distributions. We also show the insensitivity results and obtain formulas for performance measures such as throughput and mean number of jobs. In Section 3 we investigate the multiple-station, multiple-job class queueing networks with BAS mechanism. We describe the model first. We state the condition on the total number of jobs permitted in the network. Then we use the equivalency between blocking and nonblocking networks and obtain exact solution for equilibrium state probabilities. We also derive formulas for performance measures to compute throughput and mean queue lengths. Finally Section 4 concludes the paper.

2. Two-station BAS networks with different station types and single job class

2.1. Model description

We consider two stations, labeled 1 and 2. The stations form a cyclic network, i.e. jobs that leave station 1 go to station 2 and vice versa. We will later show that the case in which jobs are allowed to return to the station where they finish service can be reduced to this case, so there is no loss of generality in this. The capacity of station i is denoted by b_i. The service requirements at station i are exponentially distributed with rate μ_i.

We assume *a scheduling discipline* that can be described in terms of the work of [11, 12, 16], i.e. there are functions:

$f_i(k)$: the total service effort of the server if there are k jobs in station i. Clearly, $f_i(k)=0$ if $k=0$,

$\phi_i(l, k)$: the fraction of the service effort expended on the job in position l in the server when there are k jobs in station i. This requires $\sum_{1\leqslant l\leqslant k}\phi_i(l, k)=1$ and $\phi_i(l, k)=0$ if $l<0$ or $l>k$,

$\psi_i(l, k)$: the probability that a newly arrived job is put in position l of the queue of station i when there are k jobs in the queue before its arrival. This requires $\sum_{1\leqslant l\leqslant k+1}\psi_i(l, k)=1$ and $\psi_i(l, k)=0$, whenever $l<0$ or $l>k+1$.

A scheduling discipline of this class is called *symmetric* if we also have

$$\phi(l, k)=\psi(l, k+1). \tag{1}$$

It is a well-known result [16] that with scheduling disciplines as these classical (nonblocking) queueing networks have a product form solution whenever either:

- The service time distributions are arbitrary (they may even depend on the job class) and the scheduling discipline is symmetric.
- The service times of all classes have the same exponential distribution.

For the multiple-server (MS) case we write m_i for the number of places in station i which receive service. For later convenience we define

$$n_i = \min(k_i, m_i). \tag{2}$$

In terms of the above functions, we have

$$\phi_i(l, k) = \begin{cases} 0 & \text{if } k=0 \text{ or } l>n_i, \\ 1/n_i & \text{if } 1 \leqslant l \leqslant n_i. \end{cases} \tag{3}$$

The blocked jobs in service positions do not make any use of the service effort assigned to their positions in the queue so that service effort is lost.

Note that we have both the total service effort made at the station in the form of $f_i(k_i)$ and also the fraction of this service effort assigned to position l by $\phi_i(l, k_i)$. We get the three cases we consider in this paper by taking m_i as 1 for FCFS and as b_i for processor sharing (PS). The definitions of MS and PS are slightly unusual in that we will assume that for the three cases we compare the functions $f_i(\cdot)$ are the same. Otherwise, the comparison would make little sense.

We restrict the functions $\phi(\cdot,\cdot)$ considered to those that distribute the service effort evenly among the served positions of the queue, because in that way we can forget the detailed positions of the blocked jobs in the queue. Otherwise, the rates at which jobs finish service at a station is affected by the ordering of the jobs. In this way they only contribute via a binomial coefficient for their multiplicity.

As for the placement function, we assume

$$\psi_i(l, k) = 0 \quad \text{for } 1 \leqslant l \leqslant n_i. \tag{4}$$

This is to ensure that no newly arrived job can displace a blocked job from a served position in the queue. We will only make use of this fact, so the placement of the arriving jobs could even depend on the blocked jobs present, as long as no blocked job leaves a served position.

The total number of jobs in the network is K. There must be some free space in the network, since otherwise the network deadlocks immediately. We also require that there be blocking, so that

$$\min(b_1, b_2) < K < b_1 + b_2. \tag{5}$$

2.2. Global balance equations

We derive the exact distributions for the three scheduling disciplines, FCFS, PS, and MS. We are mainly interested in the mean number of jobs in each station, the mean number of blocked jobs (which we take to average the mean number of jobs that are in service positions with their service finished, but which cannot proceed since the destination is full) and the throughputs.

We can distinguish three different ranges of operation of this system:

- none of the stations is full, there is no blocking,
- station 1 is full, there might be blocked jobs in station 2,
- station 2 is full, there could be blocked jobs in station 1.

Note that two stations cannot be blocked simultaneously otherwise deadlock would occur. We assume that the network is deadlock free [3]. Clearly, the situations in which one of the stations contains blocked jobs are symmetric, so it is enough to consider one of them. We will discuss the case in which station 2 is full and station 1 may contain blocked jobs.

The state in the most general case (MS) can be described by

$$(k_1, \kappa_1; k_2, \kappa_2), \tag{6}$$

where k_i is the total number of jobs and κ_i the number of blocked jobs in station i, respectively. This description is redundant, since we have $k_2 = K - k_1$ and κ_i can only be nonzero when $k_{3-i} = b_{3-i}$. We will use because it reflects the underlying symmetry very well.

The restrictions on the possible values of the κ_i are the only difference among the policies considered. For FCFS κ_i is either 0 or 1, for MS we have $0 \leqslant \kappa_i \leqslant \min(k_i, m_i)$ and for PS $0 \leqslant \kappa_i \leqslant k_i$. In general, we have $0 \leqslant \kappa_i \leqslant n_i$, with definition (2) of n_i.

2.2.1. Balance equations for nonblocked states

For nonblocking states the global balance equations are just

$$\begin{aligned}(\mu_1 f_1(k_1) + \mu_2 f_2(k_2))\pi(k_1, 0; k_2, 0) = \mu_1 f_1(k_1 + 1)\pi(k_1 + 1, 0; k_2 - 1, 0) \\ + \mu_2 f_2(k_2 + 1)\pi(k_1 - 1, 0; k_2 + 1, 0).\end{aligned} \tag{7}$$

Writing this equation in terms of

$$g(k) = \pi(k, 0; K - k, 0), \tag{8}$$

we get

$$(\mu_1 f_1(k) + \mu_2 f_2(K - k))g(k) = \mu_1 f_1(k + 1)g(k + 1) + \mu_2 f_2(K - k + 1)g(k - 1) \tag{9}$$

for $(K - b_2) \leqslant k \leqslant b_1$.

Rearranging, we have

$$\mu_2 f_2(K - k)g(k) - \mu_1 f_1(k + 1)g(k + 1) = \mu_2 f_2(K - k + 1)g(k - 1) - \mu_1 f_1(k)g(k). \tag{10}$$

Note that both sides of (10) are the expression

$$\mu_2 f_2(K - k)g(k) - \mu_1 f_1(k + 1)g(k + 1) \tag{11}$$

evaluated for k and for $k - 1$, respectively. So this has to be a constant, call it c_1. This gives a linear difference equation for $g(\cdot)$:

$$\mu_2 f_2(K - k)g(k) - \mu_1 f_1(k + 1)g(k + 1) = c_1. \tag{12}$$

The homogeneous equation (i.e. for $c_1=0$) has the general solution

$$g(k)=c_2 \prod_{1\leqslant l\leqslant k} \frac{1}{\mu_1 f_1(l)} \prod_{1\leqslant l\leqslant K-k} \frac{1}{\mu_2 f_2(l)} \tag{13}$$

for $(K-b_2)\leqslant k\leqslant b_1$.

If this were a network without blocking, we could use the balance equations for $k=0$ to show that $c_1=0$. However, in our network the state $k=0$ is not feasible because of condition (5). We will assume for now that $c_1=0$, a proof of this will be given later. Note that distribution (13) does not depend on m_i. We will make use of this fact later.

2.2.2. Balance equations for blocked states

By the obvious symmetry of the case in which stations 1 and 2 are full, the other one possibly containing blocked jobs, it is enough to consider one of them in detail. We will consider station 2 to be full. Note that this means that station 2 does not change, since as long as station 1 contains blocked jobs, whenever a job finishes service in station 2 it moves into station 1, and a blocked job moves into its place. By the same token, the total number of jobs in station 1 does not change.

A blocked job just wastes its share of the service effort. If κ jobs are blocked, the rate at which jobs leave the station is

$$\mu_i f_i(k_i)(n_i-\kappa_i)/n_i. \tag{14}$$

This is a consequence of the form of $\phi(\cdot)$ we assumed in (3). Note that in the blocked states the number of jobs in both stations is constant, the only variable here is κ_i. Also, as only jobs that are being served can get blocked, $0\leqslant\kappa_i\leqslant n_i$, where n_i is defined by (2).

The states we are considering here are of the form $(K-b_2,\kappa_1;b_2,0)$. To make the equations more compact, we will use k_1 as a shorthand for $K-b_2$ in some of the equations that follow. The only free variable here is κ_1. So we write the balance equations in terms of:

$$h_1(\kappa)=\pi(K-b_2,\kappa;b_2,0). \tag{15}$$

This gives

$$\left(\mu_1 f_1(k_1)\frac{n_1-\kappa}{n_1}+\mu_2 f_2(k_2)\right)h_1(\kappa)$$
$$=\mu_1 f_1(k_1)\frac{n_1-\kappa+1}{n_1}h_1(\kappa-1)+\mu_2 f_2(b_2)h_1(\kappa+1). \tag{16}$$

Here the first term refers to transitions in which a job finishes service in station 1 and gets blocked, while the second term is for jobs that finish service at station 2, so they move into station 1 and a blocked job from station 1 becomes unblocked and

moves into station 2. Again, the method used to derive (10) from (9) can be applied, which gives

$$\mu_2 f_2(b_2) h_1(\kappa+1) - \mu_1 f_1(k_1) \frac{n_1-\kappa}{n_1} h_1(\kappa) = c_3. \tag{17}$$

To get c_3 we use the balance equation for the case when $\kappa = n_1$, i.e. all jobs inside station 1 are blocked. The balance equation for that case is

$$\mu_1 f_1(k_1) \frac{1}{n_1} h_1(n_1-1) = \mu_2 f_2(b_2) h_1(n_1). \tag{18}$$

This is exactly equation (17) for $\kappa = n_1$ with $c_3 = 0$. So we have the solution

$$h_1(\kappa) = \frac{c_{14}}{\kappa!} \left(\frac{1}{\mu_1 f_1(k_1)} \right)^{n_1-\kappa} \left(\frac{1}{\mu_2 f_2(b_2)} \right)^{\kappa}. \tag{19}$$

Obviously, this only makes sense for $0 \leqslant \kappa \leqslant n_1$.

2.3. Exact solution for equilibrium state probabilities

There remains the problem of determining the value of the constant c_1. For this we use the balance equations for the state in which station 2 is full but no jobs are blocked inside station 1, since this case is covered both by (10) and (17), for the cases when $k = K - b_2$ and $\kappa = 0$, respectively. The value of $g(K-b_2)$ is the same as the value of $h_1(0)$, since both are the probabilities of the same physical situation.

For the special case $k = K - b_2 - 1$, equation (10) is

$$\mu_2 f_2(b_2-1) g(k_1+1) - \mu_1 f_1(k_1) g(k_1) = c_1. \tag{20}$$

The balance around the state in which station 2 is full, but there are no blocked jobs inside station 1 is

$$(\mu_1 f_1(k_1) + \mu_2 f_2(b_2)) g(k_1) = \mu_2 f_2(b_2) h_1(1) + \mu_1 f_1(k_1+1) g(k_1+1). \tag{21}$$

By solution (19) we have the following relation between $h_1(0)$ and $h_1(1)$:

$$h_1(1) = h_1(0) \frac{\mu_1 f_1(k_1)}{\mu_2 f_2(b_2)}, \tag{22}$$

which together with (21) and the equality $g(k_1) = h_1(0)$ noted above gives

$$(\mu_1 f_1(k_1) + \mu_2 f_2(b_2)) g(k_1) = \mu_1 f_1(k_1) g(k_1) + \mu_1 f_1(k_1+1) g(k_1+1), \tag{23}$$

which in turn reduces to

$$\mu_2 f_2(b_2) g(k_1) = \mu_1 f_1(k_1+1) g(k_1+1), \tag{24}$$

which is exactly equation (20) (with $c_1 = 0$). This proves the claim made in Section 3.2, so (13) is really the solution we are seeking.

The (three-piece) solution we have now involves three arbitrary constants, namely c_2, c_{14} and c_{24}, which corresponds to c_{14} but for station 2.

Since they are used very often in what follows, we define the constants N_1 and N_2 by

$$N_1 = \min(m_1, K - b_2), \tag{25}$$

$$N_2 = \min(m_2, K - b_1). \tag{26}$$

These are the values of n_1 and n_2 for blocked stations.

Making use of the fact that $g(k_1) = h_1(0)$, we can find the relationship between the two constants c_2 and c_{14} using equations (13) and (19). Equating both expressions and simplifying, we have

$$c_2 \prod_{1 \leq l \leq K - b_2} \frac{1}{\mu_1 f_1(l)} \prod_{1 \leq l \leq b_2} \frac{1}{\mu_2 f_2(l)} = c_{14} \left(\frac{1}{\mu_1 f_1(K - b_2)} \right)^{N_1}. \tag{27}$$

On the other hand, repeating the same procedure for station 2 will give us another constant c_{24}, for which we again have the equivalent of (27):

$$c_2 \prod_{1 \leq l \leq b_1} \frac{1}{\mu_1 f_1(l)} \prod_{1 \leq l \leq K - b_1} \frac{1}{\mu_2 f_2(l)} = c_{24} \left(\frac{1}{\mu_2 f_2(K - b_1)} \right)^{N_2}. \tag{28}$$

The single remaining independent constant is determined by the fact that the sum of the probabilities of all states has to be one.

Combining together the results of Sections 3.2 and 3.3, and using the value of c_2 given by (27) the complete distribution is given by

$$\pi(k_1, \kappa_1; k_2, \kappa_2) = \begin{cases} c_2 \prod_{1 \leq l \leq k} \frac{1}{\mu_1 f_1(l)} \prod_{1 \leq l \leq K - k} \frac{1}{\mu_2 f_2(l)}, \\ \frac{c_{14}}{\kappa_1!} \left(\frac{1}{\mu_1 f_1(K - b_2)} \right)^{N_1 - \kappa_1} \left(\frac{1}{\mu_2 f_2(b_2)} \right)^{\kappa_1}, \\ \frac{c_{24}}{\kappa_2!} \left(\frac{1}{\mu_1 f_1(b_1)} \right)^{\kappa_2} \left(\frac{1}{\mu_2 f_2(K - b_1)} \right)^{N_2 - \kappa_2}. \end{cases} \tag{29}$$

The first case is for nonblocked states, i.e. states of the form $(k_1, 0; k_2, 0)$; the second case is for states in which jobs in station 1 are being blocked by station 2, i.e. of the form $(K - b_2, \kappa_1; b_2, 0)$ with $0 \leq \kappa_1 \leq N_1$; and the third case is the symmetrical of the second, states of the form $(b_1, 0; K - b_1, \kappa_2)$ with $0 \leq \kappa_2 \leq N_2$. Note that the "blocked" states with $\kappa = 0$ are really the same as the states in which the stations are full.

2.4. Recirculation

The case considered so far is the case of a cyclic network, where jobs go to station 2 after finishing service at station 1 and vice versa. In general, recirculation (jobs that return to their origin station immediately) cannot be allowed because it can give rise

to deadlock. However, in the case where the "recirculating" station has infinite capacity, no deadlock can arise. This is the case when, for example,

$$b_2 < K < b_1. \tag{30}$$

In this case jobs can never get blocked inside station 2, since station 1 can never be full. We can then assume that the probability that a job leaving station 1 wants to return to station 1 is nonzero. Calling this value α, the complete set of values for the p_{ij} is as follows:

$$\begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} \alpha & 1-\alpha \\ 1 & 0 \end{pmatrix}. \tag{31}$$

Note that when a nonblocked job finishes service at station 1 and returns to it, the state of the network does not really change. The balance equations (7) and (16) are then modified only in that the rate at which jobs leave station 1 is multiplied by $(1-\alpha)$. So, the network with recirculation around station 1 is governed by the same balance equations as the network we have been considering, just with

$$\hat{\mu}_1 = (1-\alpha)\mu_1. \tag{32}$$

Also, the throughput of station 1 increases:

$$\hat{\lambda}_1 = \frac{\lambda_1}{1-\alpha}. \tag{33}$$

Except for that, the performance measures of the network with recirculation are the same as in the network without, just with the corrected μ_1.

2.5. Insensitivity

One interesting property of classical networks is the so-called *insensitivity*, which means that the probabilities of the states (and consequently of the performance measures) do not depend on the scheduling discipline nor on the distribution of the service time distributions when the network has a product form solution. For *repetitive-service-blocking*, where a job that cannot enter a full station returns to its original station and gets another round of service there, insensitivity has been proved by van Dijk and Tijms [23] for two-station networks like ours. The results of Akyildiz and von Brand [5, 6] also imply insensitivity for this repetitive-service-blocking mechanism as long as routing is reversible, as it is for cyclic two-station networks. So there is some hope of getting insensitivity in this case also. Besides, based on our extensive simulations, we conjectured that there is insensitivity for networks with BAS mechanism when different scheduling disciplines are used. However, the distributions for this simple case are different. We will now show that the mean number of jobs and throughput depend on the scheduling discipline in our model.

To simplify the derivations, we consider the symmetric case in which

$$\mu_1 = \mu_2, \tag{34}$$

$$b_1 = b_2. \tag{35}$$

We will furthermore assume that

$$f_1(l) = f_2(l) = \begin{cases} 0 & \text{if } l = 0, \\ 1 & \text{if } l > 0. \end{cases} \tag{36}$$

From now on we will drop the subscripts on these quantities.

2.5.1. Equilibrium state probabilities

For this case distribution (13) takes a particularly simple form. We will continue to use the functions $g(\cdot)$ and $h_i(\cdot)$ defined by (8) and (15), respectively. We have

$$c_{14} = c_2 \mu^{N_1 - K}, \tag{37}$$

$$c_{24} = c_2 \mu^{N_2 - K}, \tag{38}$$

so the distribution turns out to be:

$$g(k) = c_2 \mu^{-K}, \tag{39}$$

$$h_i(\kappa) = \frac{c_2 \mu^{-K}}{\kappa!}. \tag{40}$$

The constant c_2 is determined from

$$1 = \sum_{K-b \leqslant k \leqslant b} g(k) + \sum_{1 \leqslant \kappa_1 \leqslant N_1} h_1(\kappa_1) + \sum_{1 \leqslant \kappa_2 \leqslant N_2} h_2(\kappa_2). \tag{41}$$

The cases $\kappa = 0$ are omitted in the second and third sums since in those states no job is blocked, and so this case is included in the first summation. Substituting (39) and (40) into (41) we get

$$1 = c_2((b - (K - b) + 1)\mu^{-K} + \mu^{-K} S(N_1) + \mu^{-K} S(N_2)). \tag{42}$$

Here we used the shorthand

$$S(N) = \sum_{1 \leqslant k \leqslant N} \frac{1}{k!}. \tag{43}$$

Note that this is close to the sum that defines e:

$$e = \sum_{0 \leqslant k < \infty} \frac{1}{k!}. \tag{44}$$

This sum converges very rapidly, as is clear from the factorial in the denominator. Rearranging (42), we have

$$c_2 = \mu^K (2b - K + 1 + S(N_1) + S(N_2)). \tag{45}$$

2.5.2. Mean number of jobs

Now we are ready to compute the mean number of jobs in each station. The mean number of jobs inside station i is

$$\bar{k}_i = c_2 \mu^{-K} \left(\sum_{K-b \leqslant k_i \leqslant b} k_i + (K-b)S(N_1) + bS(N_2) \right). \tag{46}$$

The second term is the probability that there are blocked jobs inside station 1, in which case there are $K-b$ jobs in it. Similarly, the third term is for the case where there are blocked jobs inside station 2, in which case station 1 is full.

It is clear from (46) that if the values of the N_i are different for the stations, the mean number of jobs in them will be different. However, given our symmetry assumptions, the only difference between the stations is the scheduling discipline. Note also that the difference will normally be very small, since it is essentially a part of the sum in (44).

2.5.3. Throughput

Even though the mean number of jobs is affected by the scheduling discipline, it is conceivable that the throughput is not. We now show that the throughput also depends on the scheduling discipline. As the throughput is exactly the same in both stations, we will consider the case in which $N_1 = N_2$, in addition to the symmetry requirements we imposed in equations (34)–(36). We will then show that throughput depends on their common value.

The throughput is essentially the mean number of service positions that are active (i.e. occupied by a job that is receiving service) in any one of the stations, say station 1. This is given by

$$\lambda = \mu^{-K+1} c_2 \left(\sum_{K-b \leqslant k \leqslant m} k + \sum_{m < k \leqslant b} m + \sum_{1 \leqslant k \leqslant K-b} \frac{K-b-\kappa}{\kappa!} + mS(m) \right). \tag{47}$$

Here the first term corresponds to the case where not all service positions in the station are occupied, while the second term is still for the case where the station itself is not full (or is full but does not block any jobs) but all the service positions are occupied, where we are arbitrarily assuming that the number of service positions is greater than the minimal number of jobs in the station. The third term is for the case in which the blocked jobs occupy service positions and the fourth term considers the case in which the station is full, blocking jobs in the other station.

Rearranging equation (47) and expressing the third term in terms of $S(\cdot)$, we have

$$\lambda = c_2 \mu^{-K+1} \left(\frac{(2b+1)m - m^2}{2} + mS(m) - \frac{(K-b-1)(K-b)}{2} \right.$$
$$\left. + (K-b)S(K-b) - S(K-b-1) - 1 \right). \tag{48}$$

Clearly this depends on m, so the throughput of this system depends on the scheduling discipline.

Remark. Unfortunately we were not able to extend these results to multiple-job classes. However, as we will show in the next section we were able to find exact solutions for queueing networks with BAS mechanism, more than two stations and different station types. However, we have a condition on the total number of jobs allowed in the network.

3. BAS queueing networks with multiple-job classes and a limited number of jobs

3.1. Model description

Here we consider queueing networks formed by *stations*, numbered $i=1,2,3,\ldots,N$. The capacity (i.e. including buffer space and the spaces in the servers) of station i is denoted b_i. We assume multiple *classes* of jobs, labeled with lowercase Greek letters ($\alpha, \beta, \ldots$). The probability that a job of class α leaving station i wants to join station j in class β is written as $p_{i\alpha,j\beta}$. We will define the *relative throughputs* $e_{i\alpha}$ as any solution of the homogeneous system

$$e_{j\beta}=\sum_{i\alpha} p_{i\alpha,j\beta}e_{i\alpha}. \tag{49}$$

The name given to these quantities will be justified when we discuss performance measures for the network. The set of pairs (station, class) that a particular job may enter by the above is called a *routing chain*. We assume that the Markov chain represented by each routing chain of the network is irreducible. The routing chain that contains (i,α) will be denoted $\mathscr{R}_{i\alpha}$, while $\mathscr{R}_i$ is the set of routing chains which pass through station i. We will use $r,s,\ldots$ as indices ranging over routing chains.

We will denote the total number of jobs in the buffer of station i by k_i and the total number of jobs in the buffer of station i belonging to routing chain r by k_{ir}. The total number of jobs at station i (inside the station and waiting in other stations upstream for a place in the station) is denoted by κ_i; the corresponding numbers for class α and routing chain r are $\kappa_{i\alpha}$ and κ_{ir}, respectively. The total number of jobs in routing chain r is given by K_r. The total number of jobs in the network is K.

We assume that the scheduling discipline is such that the corresponding classical network has a product form solution, i.e. there are same functions as defined in Section 2, viz, $f_i(k), \phi_i(l,k)$ and $\psi_i(l,\kappa)$ with the difference in case of the latter where blocked jobs (i.e. jobs that are waiting inside other stations to enter station i) are assumed to continue to the queue outside of the station, so we do not require $\psi(l,\kappa)$ to vanish for $l>b_i$. To keep jobs outside of the station separate from the jobs inside the station, we require $\psi(l,\kappa)=0$ when $l\leqslant b_i$ and $\kappa>b_i$.

By the requirements we place on the scheduling discipline because of blocking, it cannot be symmetric. To get a product form solution for the general case based on considering the job waiting outside of the station as using a "shadow" space in the buffer, we must assume exponential service time distributions that are identical for all job classes in those stations that may block. Stations whose size is effectively infinite (because they can be full without any jobs waiting outside) are exempted from this requirement.

3.2. Equivalence between blocking and nonblocking networks

We set up the balance equations for the blocking network we described above under the assumption that if a station is full, there is at most one job outside that station that may try to enter it. This simplifies the description of the state, since it is then impossible for a real queue of jobs to form outside of the station waiting to enter. In essence, what happens if a job is blocked is that the station in which it is blocked becomes functionally just another space in the buffer of the blocking station. However, if the blocking station had one more space in it, there would never be blocking and the network is simply a classical network. This was used by Onvural and Perros [21, 22] to derive an exact solution for queueing networks with BAS and one-job class. Here we extend their results to multiple-job classes and give a rigorous derivation for the result.

The basic condition is that whenever station i is full, there can be at most one job outside that station in one of the routing chains that visit that station. Considering station i, this gives rise to the following:

$$\sum_{r \in \mathscr{R}_i} K_r \leqslant b_i + 1. \tag{50}$$

Furthermore, the job that tries to enter station i from station j has to be alone in station j, since otherwise it will affect the other jobs in it. So, all stations j that feed a station i for which relation (50) is satisfied as an equality can only be visited by the routing chains that visit station i.

We can describe the state of a station in the network under the present assumptions by the vector of the states of the stations:

$$S = (s_1, s_2, \ldots, s_N), \tag{51}$$

where the state of each individual station i is given by

$$s_i = \begin{cases} \langle c_{i1}, c_{i2}, \ldots, c_{ik_i} \rangle & \text{nonblocking states,} \\ \langle c_{i1}, c_{i2}, \ldots, c_{ib_i}, (j, c_{j1}^*) \rangle & \text{when blocking station } j. \end{cases} \tag{52}$$

Here we distinguished the class of the blocked job, since it is the class the job acquires *after* finishing service and deciding to go to station i. Note also that the station from which the job comes has been recorded in the state.

We will call the states corresponding to S and s_i in (51) and (52) for the nonblocking network one gets by waiving the capacity restrictions S^+ and s_i^+, respectively. Note that many blocked states correspond to the same state of the nonblocking network.

We define the operator $T_{il\alpha,jm\beta}(S)$ applied to state S of the network as producing the state that results when the job in position l of the queue of station i (if it is of class α) is placed in position m of station j in class β. We will also use the inverse operator $T^{-1}_{il\alpha,jm\beta}$. Whenever the operation does not make sense both operators return an impossible state (a state with probability zero). Note that in the case that station j is full under the restrictions we have imposed $T_{il\alpha,jm\beta}$ just announces that the job will enter station j in class β when a space becomes available in the blocking network. The operators can be applied the same way in the corresponding nonblocking network.

3.3 Global balance equations

We will set up the global balance equations for this system by considering first the effect of a job of a particular class at a given position in a station. We get the global balance equations by summing over all possible classes, positions and stations. To keep notation simple, when we refer to numbers of jobs or the class of a particular job we mean the values for state S.

The intensity at which our system enters state S due to a job in class α at position l of station i is given by

$$\sum_{jm\beta} p_{j\beta,i\alpha}\phi_j(m,k_j+1)\mu_{j\beta}f_j(m,k_j+1)\psi_i(l,\kappa_i-1)\pi(T^{-1}_{jm\beta,il\alpha}(S)). \tag{53}$$

The intensity at which the system leaves the same state due to this job is

$$\sum_{jm\beta} p_{i\alpha,j\beta}\phi_i(l,k_i)\mu_{i\alpha}f_i(l,k_i)\psi_j(m,\kappa_j)\pi(S). \tag{54}$$

Note that the system cannot leave a state due to a blocked job, so *job local balance* [12, 16], which would equate (53) with (54) cannot possibly be satisfied. Neither does summing out over the possible job classes help, which would give *local balance* [11], we have to consider the global balance equations we get by summing out over all possible classes and positions. Summing (53) and (54) over l and α, rearranging and taking factors out of the sums whenever possible we get the global balance equations:

$$\sum_{il\alpha}\psi_i(l,\kappa_i-1)\sum_{jm\beta}p_{j\beta,i\alpha}\phi_j(m,k_j+1)\mu_{j\beta}f_j(m,k_j+1)\pi(T^{-1}_{jm\beta,il\alpha}(S))$$

$$=\pi(S)\sum_{il\alpha}\phi_i(l,k_i)\mu_{i\alpha}f_i(l,k_i)\sum_{jm\beta}p_{i\alpha,j\beta}\psi_j(m,\kappa_j). \tag{55}$$

We have lumped all states in which a station (with a given configuration of its queue) blocks a particular job, regardless of the station upstream in which that job is. The global balance equations that result have the same form as the corresponding

equations for the classical network one gets when the restrictions on station capacities are dropped. This means that both networks have the same state space (up to the lumping of blocked states) and the same state probabilities.

Actually, we have proved an equivalence. If the nonblocking network has a product form solution, so has the blocking network. The nonblocking network, given the fact pointed out when we discussed them that the scheduling discipline is *not* symmetric at blocking stations (which satisfy (50) with equality), a product form solution is given only when all jobs have the same exponential service time distribution there. Other stations underly the normal restrictions for product form [11] since the corresponding balance equations are in no way affected. This is true even for the stations in which there could be blocked jobs.

3.4. Equilibrium probabilities for the blocked states

The above method only gives the probability that a job is blocked trying to enter station i, it does not give the probability that the job is blocked inside a particular upstream station. We now derive the requisite probabilities.

For a blocked state S all "neighboring" states, i.e. states from which the blocked state is entered and which are entered from the blocked state are all "classical" states without blocking. So, their probabilities are given by the expressions for the corresponding classical network. We can then set up the balance equation for entering and leaving the blocked state of interest, which gives its probability.

For the network without blocking (assuming that all service time distributions are exponential for simplicity) the probability of state S^+ is given by

$$\pi(S^+)=\frac{1}{G}\prod_i\left[\prod_{1\leqslant l\leqslant k_i}\frac{1}{f_i(l)}\prod_\alpha\left(\frac{e_{i\alpha}}{\mu_{i\alpha}}\right)^{k_{i\alpha}}\right]. \tag{56}$$

Here G is a normalization constant chosen so that the probabilities add up to one.

For blocked state $S=\langle c_{i1}, c_{i2}, \ldots, c_{ib_i}, (j\alpha)\rangle$ we are interested in the neighboring state in which the blocked job has not finished service yet. This state is $S_\beta = T^{-1}_{j1\beta, ib_i+1\alpha}(S)$, whose probability can be expressed in terms of state S^+ of the nonblocking network in which the blocked job is at the end of the queue of station i:

$$\pi(S_\beta)=\frac{\mu_i f_i(b_i+1)}{e_{i\alpha}}\frac{e_{j\beta}p_{j\beta,i\alpha}}{\mu_{j\beta}f_j(1)}\pi(S^+). \tag{57}$$

The rate at which state S is left is simply the rate at which jobs finish service in station i, i.e. $\mu_i f_i(b_i)$. So we have the balance equation around state S:

$$\mu_i f_i(b_i)\pi(S)=\sum_\beta p_{j\beta,i\alpha}\mu_{j\beta}f_j(1)\pi(S_\beta). \tag{58}$$

Using the expression (56) for the probability of state S^+, the probability of state S_β (a "classical" state, since the job is not yet blocked) can be written:

$$\pi(S_\beta)=\frac{\mu_i f_i(b_i+1)}{e_{i\alpha}}\frac{e_{j\beta}}{\mu_{j\beta} f_j(1)}\pi(S^+). \tag{59}$$

Substituting (59) into (58), we get

$$\mu_i f_i(b_i+1)\pi(S)=\frac{\mu_i f_i(b_i)}{e_{i\alpha}}\pi(S^+)\sum_\beta e_{j\beta}p_{j\beta,i\alpha}, \tag{60}$$

from which we get, since $f_i(b_i+1)=f_i(b_i)$:

$$\pi(S)=\pi(S^+)\sum_\beta e_{j\beta}p_{j\beta,i\alpha}. \tag{61}$$

Note that this probability will usually depend on the class considered. Also, by the product form of $\pi(S^+)$ we can write (61) as a factor for station j, namely

$$\frac{e_{j\beta}}{\mu_j}. \tag{62}$$

Usually we will be interested not in this kind of detailed state (which describes the exact order of the jobs in the station) but in the state one gets for a given vector of numbers of jobs in each class. This will multiply both $\pi(S)$ and $\pi(S^+)$ in equation (61) by the same multinomial coefficient, since the queue of station i is exactly the same for both.

By the restriction on the number of jobs of Section 3.1, these are the only possibilities of a job of class α being blocked in station j by station i, except for other jobs in the rest of the network. By the product form of the solution of the network, when we sum out over all the states that have this particular configuration in stations i and j, we get a factor that is just the normalizing constant for the rest of the network, without stations i and j and without the jobs in the routing chains $\mathscr{R}_i$. This means that, in a sense, we can analyze this case in isolation.

The probability of a job in class α blocked trying to enter station i is just the probability given by (61) multiplied by the appropriate multinominal coefficient. Equation (61) considers job *classes*, so this has to be summed over all possible class memberships of the jobs inside station i. As in the classical case, the sum is nothing but the expansion of the sum of the class throughputs to the power K_r each (except for the chain which contains $i\alpha$, for which it is one less) multiplied by a constant. This allows us to write probability (61) in terms of routing chains, not classes. Using $\boldsymbol{k}_i$ to represent the vector of numbers of jobs in each routing chain in station i, we define

$$A_i(\boldsymbol{k}_i)=\prod_{1\leqslant l\leqslant k_i}\frac{1}{f_i(l)}\prod_r\left(\frac{e_{ir}}{\mu_i}\right)^{k_{ir}}. \tag{63}$$

Just the part of state S^+ that considers the last job in the queue (the blocked job) has to consider job classes. This leads to

$$\mathscr{P}(\alpha \text{ blocked in } j \text{ by } i)=\frac{1}{G}\prod_{m\neq i,j} A_m(\boldsymbol{k}_m)\, A_i(\boldsymbol{k}_i)\sum_{\beta}\frac{p_{j\beta,i\alpha}e_{j\beta}\,e_{i\alpha}}{\mu_{i\alpha}f_i(b_i)}. \tag{64}$$

To get the probability of a job in routing chain r blocked inside station j, the above probability is summed over all classes α in chain r such that for some class β in station j, we have

$$p_{j\beta,i\alpha}\neq 0. \tag{65}$$

To get the probability that a job of routing chain $\mathscr{R}$ is blocked *inside* station i is to add the above over the possible destinations of such a job.

3.5. Performance measures

As the probabilities of the states of the blocking network are the same as in the nonblocking network, we can compute certain performance measures for the network using methods for a classical network. Note that the *number of jobs* in the stations are not the same as in the corresponding classical network.

3.5.1. Throughputs

The throughput of each station for a job class can be expressed as the rate at which jobs of the class under consideration finish service at the station. This is clearly the same in the blocking network as in the corresponding network without blocking. The only difference is that in the nonblocking case the jobs move out of the station in the moment their service finishes, while a blocked job stays some additional time in the station. However, it eventually moves out, before any other job in the station. So the throughputs are the same in both cases.

In the corresponding classical network, the $e_{i\alpha}$ are relative throughputs, in that

$$\frac{e_{i\alpha}}{e_{j\beta}}=\frac{\lambda_{i\alpha}}{\lambda_{j\beta}} \tag{66}$$

for all $i\alpha, j\beta$. As the throughputs of the blocking network and the corresponding classical network are the same, this relation also holds in the blocking network, thus justifying the name "relative throughputs" (Section 3.1).

3.5.2. Average queue lengths

The average queue lengths of the blocking and the corresponding classical network are different, since in the classical network the queue may extend to length b_i+1, and the blocked jobs are not considered to be inside the stations in which they are blocked.

We will write $\bar{k}_{ir}$ for the mean number of jobs of routing chain r waiting for service at station i (this is the value that algorithms for the classical network give for the mean

number of jobs in the station) and $\bar{q}_{ir}$ for the corresponding mean number of jobs inside the station. Then we clearly have

$$\bar{q}_{ir}=\bar{k}_{ir}-\mathscr{P}(\text{a job in } r \text{ blocked at } i)+\mathscr{P}(\text{a job in } r \text{ blocked in } i). \tag{67}$$

The probabilities in (67) can be computed using the method outlined in Section 3.1.

3.5.3. Mean time in the station

The time a job stays for service at a particular station is the same for both networks. However, in the blocking network this includes time spent outside of the station waiting for a free place. This is not the same as the time the job stays in the station, since the job also stays in the station after its service is finished waiting for available space downstream.

The mean time a job waits for service is just the mean time in the station for the classical network. The mean time in the station is related to the mean queue length by Little's law,

$$\bar{\lambda}_{ir}\bar{t}_{ir}=\bar{q}_{ir}, \tag{68}$$

from which we can easily get $\bar{t}_{ir}$, since both $\bar{\lambda}_{ir}$ and $\bar{q}_{ir}$ can be computed for the blocking network, $\bar{\lambda}_{ir}$ being just the value of the corresponding classical network and $\bar{q}_{ir}$ being obtained from (67).

4. Conclusions

Note that several variants of BAS are possible. One possibility is that the blocked job just stays in the served position in the queue of the station, denying its use to other jobs and the service effort destined to that position being lost, or the server could shut down completely when there is a blocked job in the station. We showed in Section 2.2 that the distribution of the states in which no job is blocked does not depend on the number of service positions. Only the distribution of the blocked states does. So, with the above solution we can consider both alternatives by considering the case we have called FCFS (one service position) for the case where the server shuts down as soon as there is a blocked job inside the station.

The result for FCFS could also be derived using the equivalence between blocking-before and blocking-after-service noted in [7, 19] and the method of Gordon and Newell [14]. We did not use that method here so as to show the fundamental similarity between the three cases discussed. The method used gives the complete probability distribution for this simple case, so we could determine closed-form expressions for the mean number of jobs and the throughput for a symmetric case. This shows that for systems with BAS neither the mean number of jobs nor the throughput are independent of the scheduling discipline of the stations. This is one of the basic properties of the solutions for classical networks, so this reflects the greater complexity of networks with blocking.

We have shown that if whenever a station is full there is at most one job outside that station that may try to enter it, the equilibrium state probabilities of a queueing network with BAS have a simple relation to the corresponding probabilities for the same network without blocking. This condition is easy to check. Some performance measures for the blocking network can be computed directly by methods for classical networks. This is the case for throughput and the mean number of jobs waiting for service at a particular station.

The equivalence allows to compute the full equilibrium distribution for the queueing network with blocking. Using the distribution we deduce methods to compute some other performance measures (mean number of jobs and probabilities of blocking, that here are also the mean number of blocked jobs) for the blocking network based on values that are given by the classical algorithms for queueing networks. The algorithms are simple to implement using a convolution method to solve the corresponding nonblocking network.

References

[1] I.F. Akyildiz, Exact product form solution for queueing networks with blocking, *IEEE Trans. Comput.* **C36**(1) (1987) 122–125.

[2] I.F. Akyildiz, On the exact and approximate throughput analysis of closed queueing networks with blocking, *IEEE Trans. Software Engrg.* **SE14**(1) (1988) 62–71.

[3] I.F. Akyildiz and J. Liebeherr, Optimal deadlock free buffer allocation in multiple chain blocking networks of queues, *Proc. Internat. Conf. on the Performance of Distributed Systems and Integrated Communication Networks* (1991).

[4] I.F. Akyildiz and H.G. Perros, Special issue on queueing networks with finite capacity queues: introduction, *Performance Evaluation*, **10** (3) 1989.

[5] I.F. Akyildiz and H. von Brand, Exact solutions for open, closed and mixed queueing networks with rejection blocking, *Theoret. Comput. Sci.* **64** (1989) 203–219.

[6] I.F. Akyildiz and H. von Brand, Computational algorithms for networks of queues with rejection blocking, *Acta Inform.* **26** (1989) 559–576.

[7] S. Balsamo, V. de Nitto Persone and G. Iazeoalla, Identity and reducibility properties of some blocking and nonblocking mechanisms in congested networks, *Flow Control of Congested Networks*, NATO ASI Series (Springer, Berlin, 1987) 243–254.

[8] S. Balsamo and L. Donatiello, On the cycle time distribution in a two-stage cyclic network with blocking, *IEEE Trans. Software Engrg.* **SE15** (10) (1989) 1205–1216.

[9] F. Baskett, K.M. Chandy, R.R. Muntz and F.G. Palacios, Open, closed and mixed networks of queues with different classes of customers, *J. ACM* **22**(15) (1975) 248–260.

[10] P.P. Bocharov, On the two-node queueing networks with finite capacity, *Proc. 1st Internat. Workshop on Queueing Networks with Blocking*, (North Holland, Amsterdam, 1989) 105–125.

[11] K.M. Chandy, J.H. Howard and D.F. Towsley, Product form and local balance in queueing networks, *J. ACM* **24**(2) (1977) 250–263.

[12] K.M. Chandy and A.J. Martin, A characterization of product form queueing networks, *J. ACM* **30** (1983) 286–299.

[13] A. Conway and N. Georganas, *Queueing Networks: Exact Computational Algorithms* (MIT Press, Cambridge, MA, 1989).

[14] W.J. Gordon and G.F. Newell, Cyclic queueing systems with restricted queues, *Oper. Res.* **15** (1967) 266–277.

[15] F.P. Kelly, Networks of queues with customers of different types, *J. Appl. Probab.* **12** (1975) 542–554.

[16] F.P. Kelly, Networks of queues, *Adv. in Appl. Probab.* **8**(2) (1976) 416–432.

[17] R.O. Onvural, A note on the product form solutions of multiclass closed queueing networks with blocking, *Performance Evaluation* **10**(3) (1989) 247–255.
[18] R.O. Onvural, A survey of closed queueing networks with finite buffers, *ACM Comput. Surveys* **22**(2) (1990) 83–121.
[19] R.O. Onvural and H.G. Perros, On equivalencies of blocking mechanisms in queueing networks with blocking, *Oper. Res. Lett.* **5** (1986) 293–297.
[20] R.O. Onvural and H.G. Perros, Some equivalencies between closed queueing networks with blocking, *Performance Evaluation* **9** (1988) 111–118.
[21] R. Onvural and H.G. Perros, Equivalences between open and closed queueing networks with finite buffers, *Performance Evaluation* **9** (1988/1989) 263–269.
[22] R.O. Onvural and H.G. Perros, Throughput analysis of cyclic queueing networks with finite buffers, *IEEE Trans. Software Engrg.* **SE15** (1989) 800–808.
[23] N. van Dijk and H. Tijms, Insensitivity in two-node blocking models with applications, *Proc. Teletraffic Analysis Computer Performance Evaluation* (North Holland, Amsterdam, 1986). 329–340.