Central Server Models with Multiple Job Classes, State Dependent Routing, and Rejection Blocking

I. F. AKYILDIZ, SENIOR MEMBER, IEEE, AND HORST VON BRAND

Abstract—**Central server models with multiple job classes, state dependent routing, and rejection blocking are investigated. The service time distributions may depend on the job class. Using the concept of job local balance, we prove that the equilibrium state probabilities of these networks have an exact product form solution. From the equilibrium state probabilities we deduce formulas for the throughputs. An algorithm to compute performance measures, like the throughputs and the mean number of jobs is given. The complexity of the algorithm is discussed.**

Index Terms—**Blocking, finite station capacities, performance evaluation, performance measures, queueing network models.**

I. Introduction

CENTRAL Server Models are important because most computer systems can be modeled as central server models either directly or after replacing the I/O subsystems by equivalent stations. A product form solution for a central server model with state dependent routing and a single job class was first derived by Towsley [28]. General service time distributions are allowed at a station if the scheduling discipline is symmetric. Sauer [25] obtains mean queue lengths, throughputs and the marginal distributions of the station populations for the same model. Yao and Buzacott [29] extend the central server model in [28] to multiple job classes. The service time distributions are assumed to be exponential and the same for all job classes at a station. The jobs are serviced in random order [27]. There is no blocking, capacity restrictions at the stations are enforced by the routing probabilities. More general queueing networks with state dependent routing are studied by Serfozo [26]. The model of a system of flexible manufacturing cells of Dallery and Yao [12] is also a central server model with rejection blocking, but the routing probabilities are fixed, i.e., state independent.

Krzesinski [19] considers BCMP [6], Kelly [17] type queueing networks and partitions them into two subnetworks, Γ and Θ. The stations in subnetwork Θ are partitioned into disjoint groups called branches. The branches are arranged into a hierarchy of nested subnetworks. A set of state-dependent routing probabilities is used to admit jobs from subnetwork Γ into the individual branches of the subnetwork Θ. The state-dependent routing probabilities are products of linear functions of the branch and subnetwork populations such that the entire network has a product form solution. He extends the results of Towsley [28] to multiple job class queueing networks. One can easily recognize that Yao and Buzacott's work [29] described above, is of a similar nature. Krzesinski [19] claims that there exists a relationship between state-dependent routing and blocking. In other words, the blocking can be enforced by state-dependent routing probabilities. Yao and Buzacott [29] used this idea for the investigation of finite capacity queueing networks as described above.

This work extends the results of Towsley [28] and Yao and Buzacott [29] for central server models with state-dependent routing to a model with multiple job classes and rejection blocking. The difference of this work with the previous studies is that it considers both state-dependent routing probabilities and rejection blocking in a queueing network model. Additionally, the model may have different station types. Note also that we give computational algorithms for performance measures.

The blocking policy we consider in this work is the rejection blocking policy [1]–[4], [7], [11], [13]–[16], [18], [22]. In rejection blocking policy the blocking events occur when a job that finishes service at station i determines, according to the routing probabilities for its class, to which station it tries to go next. According to the blocking function for that job at its destination station, it is determined if the job is accepted. If the job is rejected, it returns to station i, where it is treated exactly like a newly arrived job. The only exception to this is that it cannot be rejected. In station i the job gets another round of service, after which it again selects a destination, possibly a different one.

In our model, each class has an independent routing strategy of the class considered by Towsley [28]. The queueing network is closed, and no class changes are al-

Manuscript received February 11, 1988; revised May 23, 1989. Recommended by D. Opferman. The work of I. F. Akyildiz was supported in part by the Air Force Office of Scientific Research under Grant AFOSR-87-0160.

I. F. Akyildiz is with the School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA 30332.

H. von Brand is with the Departamento Informatica, Universidad Tecnica Federico Santa Maria, Valparaiso, Chile.

IEEE Log Number 8930135.

lowed. Stations in the network may have general service time distributions. Service time distributions may depend on the job class provided that the scheduling discipline at the station is symmetric (station type I). For arbitrary scheduling disciplines the service time distributions at the station must be the same exponential distribution for all job classes (station type II). The routing probabilities from the central server to the peripheral stations are state dependent. Additionally, the rejection policy is allowed in the model.

One application example is a central server model of a computer system with the CPU as the central server and the remaining stations as the disk devices. State dependent routing probabilities are used to route the jobs from the CPU to the disks. Each disk has finite capacity such that a certain maximum number of jobs can be accomodated. If the capacity of each disk is full, then the job coming from the CPU will be rejected and will be sent back to the CPU where it will receive another round of service. State dependent routing offers substantial throughput gains over state independent routing [19]. State independent routing probabilities cannot prevent the routing of jobs to already congested disk stations, the state dependent routing probabilities preferentially route jobs to the least loaded disks. Another application of this model might be a computer communication network with adaptive routing and window flow control.

The paper is organized as follows. In Section I we describe the state-of-the-art. In Section II we describe the model. In Section III we derive the transition rates which are then used to prove the exact product form solution for equilibrium state probabilities. In Section IV, formulas are derived which are then used to outline an algorithm to compute performance measures such as throughputs, mean number of jobs, in Section V. In Section VI we discuss the complexity of the algorithm outlined in section V. Conclusions are given in Section VII.

II. Model Description

The *central server* is numbered 1 and the rest of the stations 2, 3, $\cdots$, N. The stations 2 to N are called peripheral stations. There are C different job classes in the network. A job of class α requests service at station i distributed as $F_{i\alpha}$ with mean $1/\mu_{i\alpha}$. By the results of Barbour [5] it is enough to establish our results for finite mixtures of Erlang distributions. The restrictions that Barbour imposes on the network are that there be no multiple transitions and that the arrival processes be independent of the state of the network. Both are satisfied in this model.

We will represent the service requirement distributions as mixtures of Erlang distributions of the following form:

$$F_{i\alpha} = \sum_t g_{i\alpha;t} E_{t\nu_{i\alpha}} \tag{1}$$

where $E_{t\nu_{i\alpha}}$ is the Erlang distribution with t phases, each with rate $\nu_{i\alpha}$. We assume that the sum in (1) is finite, but we refrain from giving the limits to keep notation simple. Equation (1) means that with probability $g_{i\alpha;t}$ a job of class α arriving at station i will have to traverse t exponential phases, each of which has rate $\nu_{i\alpha}$. This requires:

$$\sum_t g_{i\alpha;t} = 1.$$

It also implies:

$$\frac{1}{\mu_{i\alpha}} = \sum_t g_{i\alpha;t} \frac{t}{\nu_{i\alpha}}. \tag{2}$$

By renewal theory the probability that at an arbitrary instant a job with service requirement distribution $F_{i\alpha}$ still has to traverse s phases, including the one being traversed, is given by:

$$r_{i\alpha}(s) = \frac{\mu_{i\alpha}}{\nu_{i\alpha}} \sum_{t \geq s} g_{i\alpha;t}. \tag{3}$$

Note that:

$$r_{i\alpha}(1) = \frac{\mu_{i\alpha}}{\nu_{i\alpha}}.$$

Assume that we consider the job in station i at position l, and that it is in phase s of its service. Then we have:

$$\Pr(\text{this is the last phase of service}) = \frac{g_{i\alpha;s}}{\sum_{t \geq s} g_{i\alpha;t}} = \frac{\mu_{i\alpha} g_{i\alpha;s}}{\nu_{i\alpha} r_{i\alpha}(s)} \tag{4}$$

$$\Pr(\text{this is not the last phase of service}) = \frac{\sum_{t \geq s+1} g_{i\alpha;t}}{\sum_{t \geq s} g_{i\alpha;t}} = \frac{r_{i\alpha}(s+1)}{r_{i\alpha}(s)}. \tag{5}$$

The state of the network will described by (ordered) N-tuples of station states where a station i's state is denoted by:

$$((\kappa_{i1}, \sigma_{i1}), (\kappa_{i2}, \sigma_{i2}), \cdots, (\kappa_{ik_i}, \sigma_{ik_i})). \tag{6}$$

Here k_i is the number of jobs in station i, κ_{il} is the class of the job in position l of station i and σ_{il} is the number of remaining phases of service for that job. We will denote the number of jobs of class α in station i by $k_{i\alpha}$. We will use x and y to denote arbitrary states of the network. We define the *occupancy* of the network as an N-tuple of strings of job classes, where the ith string represents the classes of the jobs in station i in order. The *population* of the network gives the number of jobs of each class in each station. Occupancies and populations are defined in the obvious ways for single stations. The occupancy of the network will be denoted by $\boldsymbol{n}$, and the population by $\boldsymbol{k}$. For single stations we will use $\boldsymbol{n}_i$ and $\boldsymbol{k}_i$, respectively. The total number of jobs in class α is denoted by K_α.

A *scheduling discipline* (f, ϕ, ψ) is defined by [9], [10], [17], [18]:

$f(k)$ Total service effort when there are k jobs in the station.

$\phi(l, k)$ Fraction of the service effort destined to the job in position l when there are k jobs in the station (zero for l outside of $1 \le l \le k$). This requires:

$$\sum_{1 \le l \le k} \phi(l, k) = 1 \qquad \forall k. \tag{7}$$

$\psi(l, k)$ Probability that an arriving job is placed in position l when there are k jobs in the station (zero for l outside of $1 \le l \le k+1$). This requires:

$$\sum_{1 \le l \le k+1} \psi(l, k) = 1 \qquad \forall k. \tag{8}$$

Kelly [17] calls a scheduling discipline *symmetric* (Chandy and Martin [9] call them *station balancing*) if

$$\psi(l, k) = \phi(l, k+1). \tag{9}$$

This framework clearly does not describe all possible scheduling disciplines, for example there is no way to give one job class priority over another. Scheduling disciplines that depend on the service requirements, like Shortest Job First (SJF), cannot be described either. Nevertheless, the class of scheduling disciplines that can be described is rich. Some examples are:

FCFS First come, first served is described by $\phi(1, k) = 1$ and $\psi(k+1, k) = 1$.

LCFS Last come, first served preemptive is described by $\phi(k, k) = 1$ and $\psi(k+1, k) = 1$.

PS Processor sharing is described by $\phi(l, k) = 1/k$ and $\psi(k+1, k) = 1$.

RAND Service in random order (Spirn [27]) is described by $\phi(1, k) = 1$ and $\psi(l, k) = 1/k$ for $l \ge 2$.

Other scheduling disciplines that lead to product form in classical queueing networks, like LBPS (last batch processor sharing, Noetzel [20], can also be described [9].

It should be noted that the description of a particular scheduling discipline is not unique. For example, the description for PS given above is *not* symmetric, but if we set $\psi(l, k) = 1/(k+1)$ the discipline becomes symmetric. The only difference between the two is that this alternative does not keep the jobs in their order of arrival, while the description given above does. Of the remaining disciplines, FCFS and RAND are not symmetric, while LCFS is.

We assume that a job selects a service requirement before starting to get service, i.e., when a job enters station i in class α it is assigned a number of phases of service according to the $g_{i\alpha;s}$. If a job in class α is in position l of station i and the number of jobs in station i is k_i, the rate at which that job advances to its next phase of service (or finishes service at the station if it is in its last phase of service there) is $\nu_{i\alpha} f_i(k)\, \phi_i(l, k_i)$.

We call the probability that a job is accepted at a station the *blocking function* of the station. In the most general case, the blocking function of a station could depend on the state of the entire network. In our model (as in the models of Pittel [22], Hordijk and van Dijk [15], [16], Akyildiz and von Brand [2], [3], Cohen [11], van Dijk and Tijms [13], and van Dijk and Akyildiz [14]) we allow a dependence only on the state of the destination station. The probability that a job is accepted depends on its class.

We write the probability that a job of class α arriving at station i is accepted when there are a total of k_i jobs in it, of which $k_{i\alpha}$ are of class α as:

$$b_{i\alpha}(k_i) = h_{i\alpha}(k_{i\alpha})\, h_i(k_i). \tag{10}$$

Here $h_{i\alpha}$ and h_i are arbitrary functions. The only restriction on them is that if $h_i(l) = 0$ then $h_i(k) = 0$ for all $k \ge l$. A similar restriction applies to $h_{i\alpha}$. The smallest l such that $h_i(l) = 0$ is then the maximal capacity of the station for jobs and for jobs of class α, respectively. These restrictive conditions on h_i and $h_{i\alpha}$ seem to be necessary and sufficient for the irreducibility of the Markov process that represents the queueing network.

The routing probabilities are assumed to take the form [28]:

$$p_{1j;\alpha}(k) = w_{j\alpha}(k_{j\alpha})\, w_\alpha(K_\alpha - k_{1\alpha}) \tag{11}$$

$$p_{j1;\alpha} = 1. \tag{12}$$

In (11) and (12) we assume $j \ne 1$. From (11) it follows that the functions $w_{j\alpha}$ and w_α must take the forms [28, Theorem 3]:

$$w_{j\alpha}(k) = c_\alpha k + d_{j\alpha} \tag{13}$$

$$w_\alpha(k) = \left[c_\alpha k + \sum_j d_{j\alpha} \right]^{-1}. \tag{14}$$

Here c_α and $d_{j\alpha}$ are constants. For later convenience, we define the functions:

$$V_{j\alpha}(k) = \prod_{1 \le l \le k} w_{j\alpha}(l-1) \tag{15}$$

$$V_\alpha(k) = \prod_{1 \le l \le k} w_\alpha(l-1). \tag{16}$$

To describe the movement of jobs in the network, we define:

$T_{il,jm;s}(x)$ Operator that transfers the job in position l of station i to position m of station j with s stages of service left. We will also need the inverse of this operator, which we will write $T^{-1}_{il,jm;s}(x)$. This inverse is unique whenever it is defined.

$A_{il}(x)$ Operator that advances the lth job in station i to the next phase of its service requirement (defined whenever $l \le k_i$ and $\sigma_{il} > 1$).

As mentioned before, the blocking policy we consider is the rejection blocking. A job that finishes service at station i determines, according to the routing probabilities for its class, to which station it tries to go next. According

to the blocking function for that job at its destination function, it is determined if the job is accepted. If the job is rejected, it returns to station i, where it is treated exactly like a newly arrived job. The only exception to this is that a job cannot be rejected when returning to station i. In station i the job gets another round of service, after which it again selects a destination, possibly a different one. Note that a rejected job does not necessarily return to the position in station i from which it comes. It might be placed in any position of station i's queue, according to ψ_i.

III. The Equilibrium State Distribution

We will divide the set of transition rates into two parts: Rates due to changes in the central server, station 1, and rates due to changes elsewhere. We do this because of the different form of the routing probabilities (11) and (12) for these two cases.

Assume that the class of the job in station 1 (the central server), position l is κ. The transition rates $q(x, y)$ for changes at the central server, station 1, can then be written:

$$q(x,y) = \begin{cases} p_{1j;\kappa}(k)\,\nu_{1\kappa}\phi_1(l,k_1)f_1(k_1) \\ \quad \cdot b_{j\kappa}(k_j)\,\psi_j(m,k_j)g_{j\kappa;s} \\ \quad \text{if } y = T_{1l,jm;s}(x) \\ \nu_{1\kappa}\phi_1(l,k_1)f_1(k_1)\,\psi_1(n,k_1-1)g_{j\kappa;s} \\ \quad \cdot \sum_{t\neq 1} p_{1t;\kappa}(k)[1-b_{j\kappa}(k_j)] \\ \quad \text{if } y = T_{1l,1n;s}(x) \\ \nu_{1\kappa}\phi_1(l,k_1)f_1(k_1) \\ \quad \text{if } y = A_{1l}(x) \\ 0 \quad \text{otherwise.} \end{cases} \tag{17}$$

The first line is for a job moving out of the station (finishing service at station 1 and accepted at its destination), the second line is for a job that finishes service but is rejected and returns to station 1 and the third line is for a job that advances to its next phase of service in station 1.

For changes at a peripheral station j we have, assuming that the class of the job at position m is κ:

$$q(x,y) = \begin{cases} \nu_{j\kappa}\phi_j(m,k_j)f_j(k_j)\cdot b_{1\kappa}(k_1)\,\psi_1(l,k_1)g_{1\kappa;s} \\ \quad \text{if } y = T_{jm,1l;s}(x) \\ \nu_{j\kappa}\phi_j(m,k_j)f_j(k_j)\,\psi_j(n,k_j-1)g_{j\kappa;s} \\ \quad \cdot [1-b_{1\kappa}(k_1)] \\ \quad \text{if } y = T_{jm,jn;s}(x) \\ \nu_{j\kappa}\phi_j(m,k_j)f_j(k_j) \\ \quad \text{if } y = A_{jm}(x) \\ 0 \quad \text{otherwise.} \end{cases} \tag{18}$$

Again, the first line is for a job moving out of the station, the second line is for a job that finishes service but is rejected and returns while the third line is for a job that completes a phase of service.

We are now ready to state our main result:

Theorem 1: In the central server model described in Section II, assume that all stations satisfy one of the following:

i) Have symmetric scheduling disciplines with general service requirement distributions that may depend on the job class. We call these stations type I. For scheduling disciplines in the class we consider the symmetry condition is necessary and sufficient if the service requirement distributions are different for different job classes or are nonexponential.

ii) Have exponential service requirement distributions that do not depend on the job class. Here the scheduling discipline is arbitrary in the class of disciplines we consider. We call these stations type II.

Then the equilibrium state probabilities have the following exact product form solution:

$$\pi(x) = \frac{1}{G(K)} \prod_{\alpha} \left[V_\alpha(K_\alpha - k_{1\alpha}) \prod_{l\neq 1} V_{l\alpha}(k_{l\alpha}) \right] \cdot \prod_i \left[\prod_{1\le l\le k_i} \frac{h_i(l-1)}{f_i(l)}\, r_{i\kappa}(\sigma_{il}) \right] \cdot \prod_{\alpha} \prod_{1\le l\le k_{i\alpha}} \frac{h_{i\alpha}(l-1)}{\mu_{i\alpha}} \Bigg]. \tag{19}$$

Proof: The equilibrium state distribution (19) is proved by substituting it into the global balance equations. To simplify this task, one can take simpler (and more detailed) balance equations that add up to the global balance equations. Such sets are the *job local balance equations* [9], [15]–[18] and the *local balance equations* [6], [10]. The job local balance equations equate the flow into a state due to changes at a particular position l in a station i due to jobs of class α to the rate out of that state due to the same kind of change. The local balance equations are the summation of the job local balance equations over all positions in the station. They equate the flow out of a state due to jobs of class α with the rate into that state due to the same kind of change. The global balance equations (Chapman–Kolmogorov equations) equate the rate of flow out of a state with the rate of flow into that state. They are the summation of the local balance equations over all stations and all job classes. We will use the job local balance equations in this proof.

Again we treat changes at station 1 separately from changes elsewhere. The rate out of state x due to the job at $(1, l)$ is given by:

$$\pi(x)\,\nu_{1\kappa}\phi_1(l,k_1)f_1(k_1). \tag{20}$$

The rate into state x due to changes at $(1, l)$ can be written:

$$ADV1 + MOV1 + REJ1 \tag{21}$$

where *ADV1*, *MOV1*, and *REJ1* are terms for finishing a phase of service, moving into station 1 from some other station and finishing service at station 1 but returning to it because of a rejection, respectively. Written out in full, they are:

$$ADV1 = \nu_{1\kappa}\phi_1(l, k_1) f_1(k_1)\, \pi\big(A_{1l}^{-1}(x)\big) \tag{22}$$

$$MOV1 = b_{1\kappa}(k_1 - u_\kappa)\, \psi_1(l, k_1 - 1) g_{1\kappa;\sigma_{1l}} \cdot \sum_{j \neq 1} \nu_{j\kappa} f_j(k_j + 1) \sum_{1 \le m \le k_j + 1} \phi_j(m, k_j + 1) \cdot \pi\big(T_{jm,1l;\sigma_{1l}}^{-1}(x)\big) \tag{23}$$

$$REJ1 = \sum_{1 \le n \le k_1} \nu_{1\kappa}\phi_1(n, k_1) f_1(k_1)\, \psi_1(l, k_1 - 1) g_{1\kappa;\sigma_{1l}} \cdot \sum_{j \neq 1} p_{1j;\kappa}(k)\big[1 - b_{j\kappa}(k_j)\big]\pi\big(T_{1n,1l;\sigma_{1l}}^{-1}(x)\big). \tag{24}$$

The equilibrium state probabilities that appear in (22)–(24) can be expressed from (19) in terms of $\pi(x)$ as follows:

$$\pi\big(A_{1l}^{-1}(x)\big) = \frac{r_{1\kappa}(\sigma_{1l} + 1)}{r_{1\kappa}(\sigma_{1l})}\, \pi(x) \tag{25}$$

$$\pi\big(T_{jm,1l;\sigma_{1l}}^{-1}(x)\big) = \frac{\mu_{1\kappa} f_1(k_1)}{b_{1\kappa}(k_1 - u_\kappa)\, r_{1\kappa}(\sigma_{1l})} \frac{b_{j\kappa}(k_j)\, r_{j\kappa}(1)}{\nu_{j\kappa} f_j(k_j + 1)} \cdot p_{1j;\kappa}(k)\, \pi(x) \tag{26}$$

$$\pi\big(T_{1n,1l;\sigma_{1l}}^{-1}(x)\big) = \frac{r_{1\kappa}(1)}{r_{1\kappa}(\sigma_{1l})}\, \pi(x). \tag{27}$$

Substituting (25)–(27) into (22)–(24) gives after simplifying:

$$ADV1 = \pi(x)\, \nu_{1\kappa}\phi_1(l, k_1) f_1(k_1) \frac{r_{1\kappa}(\sigma_{1l} + 1)}{r_{1\kappa}(\sigma_{1l})} \tag{28}$$

$$MOV1 = \pi(x) f_1(k_1)\, \psi_1(l, k_1 - 1) \frac{\mu_{1\kappa} g_{1\kappa;\sigma_{1l}}}{r_{1\kappa}(\sigma_{1l})} \cdot \sum_{i \neq 1} p_{1i;\kappa}(k)\, b_{i\kappa}(k_i) \tag{29}$$

$$REJ1 = \pi(x) f_1(k_1)\, \psi_1(l, k_1 - 1) \frac{\mu_{1\kappa} g_{1\kappa;\sigma_{1l}}}{r_{1\kappa}(\sigma_{1l})} \cdot \sum_{i \neq 1} p_{1i;\kappa}(k)\big[1 - b_{i\kappa}(k_i)\big]. \tag{30}$$

The rate into state x is then given by:

$$\pi(x) \frac{\nu_{1\kappa} f_1(k_1)}{r_{1\kappa}(\sigma_{1l})} \Bigg[\phi_1(l, k_1)\, r_{1\kappa}(\sigma_{1l} + 1) + \psi_1(l, k_1 - 1) \frac{\mu_{1\kappa} g_{1\kappa;\sigma_{1l}}}{\nu_{1\kappa}} \Bigg]. \tag{31}$$

Exactly the same can be done for the job at (j, m), where $j \neq 1$. The rate out of state x is given by:

$$\pi(x)\, \nu_{j\kappa}\phi_j(m, k_j) f_j(k_j) \tag{32}$$

while the rate into that state is written:

$$ADVJ + MOVJ + REJJ \tag{33}$$

where *ADVJ*, *MOVJ*, and *REJJ* are terms for finishing a phase of service, moving into station j from station 1 and finishing service at station j but returning to it because of a rejection, respectively. Written out in full, they are:

$$ADVJ = \nu_{j\kappa}\phi_j(m, k_j) f_j(k_j)\, \pi\big(A_{jm}^{-1}(x)\big) \tag{34}$$

$$MOVJ = b_{j\kappa}(k_j - u_\kappa)\, \psi_j(m, k_j - 1) g_{j\kappa;\sigma_{jm}} \cdot \nu_{1\kappa} f_1(k_1 + 1) \sum_{1 \le l \le k_1 + 1} \phi_1(l, k_1 + 1) \cdot p_{1j;\kappa}(k + u_{1l} - u_{jm})\, \pi\big(T_{1l,jm;\sigma_{jm}}^{-1}(x)\big) \tag{35}$$

$$REJJ = \sum_{1 \le n \le k_j} \nu_{j\kappa}\phi_j(n, k_j) f_j(k_j)\, \psi_j(m, k_j - 1) g_{j\kappa;\sigma_{jm}} \cdot \big[1 - b_{1\kappa}(k_1)\big]\pi\big(T_{jn,jm;\sigma_{jm}}^{-1}(x)\big). \tag{36}$$

Using the product form solution (19), the equilibrium state probabilities that appear in (34)–(36) can be expressed in terms of $\pi(x)$ as follows:

$$\pi\big(A_{jm}^{-1}(x)\big) = \frac{r_{j\kappa}(\sigma_{jm} + 1)}{r_{j\kappa}(\sigma_{jm})}\, \pi(x) \tag{37}$$

$$\pi\big(T_{1l,jm;\sigma_{jm}}^{-1}(x)\big) = \frac{\mu_{j\kappa} f_j(k_j)}{b_{j\kappa}(k_j - u_\kappa)\, r_{j\kappa}(\sigma_{jm})} \frac{b_{1\kappa}(k_1)\, r_{1\kappa}(1)}{\nu_{1\kappa} f_1(k_1 + 1)} \cdot \frac{1}{p_{1j;\kappa}(k + u_{1l} - u_{jm})}\, \pi(x) \tag{38}$$

$$\pi\big(T_{jn,jm;\sigma_{jm}}^{-1}(x)\big) = \frac{r_{j\kappa}(1)}{r_{j\kappa}(\sigma_{jm})}\, \pi(x). \tag{39}$$

In (38) we have used $k + u_{1\kappa} - u_{j\kappa}$ to indicate the population of the network in state $T_{1l,jm;s}^{-1}(x)$. Substituting (37)–(39) into (34)–(36) and simplifying gives:

$$ADVJ = \pi(x)\, \nu_{j\kappa}\phi_j(m, k_j) f_j(k_j) \frac{r_{j\kappa}(\sigma_{jm} + 1)}{r_{j\kappa}(\sigma_{jm})} \tag{40}$$

$$MOVJ = \pi(x) f_j(k_j)\, \psi_j(m, k_j - 1) \frac{\mu_{j\kappa} g_{j\kappa;\sigma_{jm}}}{r_{j\kappa}(\sigma_{jm})} b_{1\kappa}(k_1) \tag{41}$$

$$REJJ = \pi(x) f_j(k_j)\, \psi_j(m, k_j - 1) \frac{\mu_{j\kappa} g_{j\kappa;\sigma_{jm}}}{r_{j\kappa}(\sigma_{jm})} \cdot \big[1 - b_{1\kappa}(k_1)\big]. \tag{42}$$

The rate into state x is then given by:

$$\pi(x) \frac{\nu_{j\kappa} f_j(k_j)}{r_{j\kappa}(\sigma_{jm})} \Bigg[\phi_j(m, k_j)\, r_{j\kappa}(\sigma_{jm} + 1) + \psi_j(m, k_j - 1) \frac{\mu_{j\kappa} g_{j\kappa;\sigma_{jm}}}{\nu_{j\kappa}} \Bigg]. \tag{43}$$

Now (20) and (31) or (32) and (43) are the same if either:

i) The scheduling discipline is symmetric, in which case by (9):

$$\phi_j(m, k_j) = \psi_j(m, k_j - 1)$$

and by (4)

$$\nu_{j\kappa}(\sigma_{jm} + 1) + \frac{\mu_{j\kappa} g_{j\kappa:\sigma_{jm}}}{\nu_{j\kappa}} = r_{j\kappa}(\sigma_{jm}).$$

ii) the service time distribution is the same exponential for all classes. Then $\sigma_{jl} = 1$, $\nu_{j\kappa} = \mu_j$ and $r_{j\kappa}(\sigma_{jl}) = 1$, $r_{j\kappa}(\sigma_{jl} + 1) = 0$. Now summing (20) and (31) or (32) and (43) over l and using (7) and (8) gives the desired equalities. Note that in reality κ does depend on l, so the assumption that the service time distribution is the same for all classes is needed.

Finally, from (31) and (43) we can deduce that the expression (19) is indeed a solution under the conditions given. □

From the product form solution (19) we can obtain the distributions for occupancies and populations.

Corollary 1: The distributions of occupancies and populations are given by

$$\pi(\boldsymbol{n}) = \frac{1}{G(\boldsymbol{K})} \prod_{\alpha} \left[V_\alpha(K_\alpha - k_{1\alpha}) \prod_{l \neq 1} V_{l\alpha}(k_{l\alpha}) \right] \cdot \prod_i P_i(\boldsymbol{k}_i) \tag{44}$$

$$\pi(\boldsymbol{k}) = \frac{1}{G(\boldsymbol{K})} \prod_{\alpha} \left[V_\alpha(K_\alpha - k_{1\alpha}) \prod_{l \neq 1} V_{l\alpha}(k_{l\alpha}) \right] \cdot \prod_i A_i(\boldsymbol{k}_i) \tag{45}$$

where $G(\boldsymbol{K})$ is the normalization constant of theorem 1 and the functions P_i and A_i are defined by:

$$P_i(\boldsymbol{k}_i) = \prod_{1 \le l \le k_i} \frac{h_i(l-1)}{f_i(l)} \prod_{\alpha} \prod_{1 \le l \le k_\alpha} \frac{h_{i\alpha}(l-1)}{\mu_{i\alpha}} \tag{46}$$

$$A_i(\boldsymbol{k}_i) = \binom{k_i}{k_{i1} \cdots k_{iC}} P_i(\boldsymbol{k}_i). \tag{47}$$

Proof: Summing (19) over all possible numbers of phases of service left for each job in the network gives (44) and summing equation (44) over all permutations of jobs in each station then gives (45).

IV. Performance Measures

Here we will derive formulas for the throughput of the central server model described in Section II.

Theorem 2: The throughput of station j ($j \neq 1$) for jobs of class α can be written:

$$\lambda_{j\alpha}(\boldsymbol{K}) = \frac{H_{j\alpha}(\boldsymbol{K} - \boldsymbol{u}_\alpha)}{G(\boldsymbol{K})} \tag{48}$$

and the throughput of station 1 for jobs of class α is

$$\lambda_{1\alpha}(\boldsymbol{K}) = \frac{1}{G(\boldsymbol{K})} \sum_{j \neq 1} H_{j\alpha}(\boldsymbol{K} - \boldsymbol{u}_\alpha) \tag{49}$$

where $G(\boldsymbol{K})$ is the normalization constant of Theorem 1 and the function $H_{j\alpha}$ is defined by:

$$H_{j\alpha}(\boldsymbol{K}) = \sum_{\boldsymbol{k} \in S(\boldsymbol{K})} \prod_{\beta} \left[V_\beta(K_\beta - k_{1\beta}) \prod_{l \neq 1} V_{l\beta}(k_{l\beta}) \right] \cdot \prod_i A_i(\boldsymbol{k}_i)\, p_{1j:\alpha}(\boldsymbol{k})\, b_{1\alpha}(\boldsymbol{k}_1)\, b_{j\alpha}(\boldsymbol{k}_j). \tag{50}$$

Here we used $S(\boldsymbol{K})$ as the set of populations of the network when the total population of the network is $\boldsymbol{K}$.

Proof: The throughput of station j ($j \neq 1$) for jobs of class α can be written:

$$\lambda_{j\alpha}(\boldsymbol{K}) = \sum_{x} \pi(x) \sum_{1 \le m \le k_j} \iota(\kappa_{jm} = \alpha)\, \iota(\sigma_{jm} = 1) \cdot r_{j\alpha}(\sigma_{jm})\, \nu_{j\alpha} \phi_j(m, k_j)\, f_j(k_j)\, b_{1\alpha}(\boldsymbol{k}_1). \tag{51}$$

In this equation we used the function ι defined so that:

$$\iota(p) = \begin{cases} 1 & \text{if } p \text{ is true} \\ 0 & \text{if } p \text{ is false.} \end{cases} \tag{52}$$

The right-hand side of (51) is the rate at which jobs of class α finish service at station j (picked out by the ι functions) and are accepted in station 1 summed over all states of the network. The expression for the equilibrium state probabilities, (19), is independent of the order of the jobs in the stations. Substituting (19) into (51) and using this fact gives:

$$\lambda_{j\alpha}(\boldsymbol{K}) = \frac{X_{j\alpha}}{G(\boldsymbol{K})} \tag{53}$$

where we have defined:

$$X_{j\alpha} = \sum_{\boldsymbol{k} \in S(\boldsymbol{K})} \prod_{\beta} \left[V_\beta(K_\beta - k_{1\beta}) \prod_{l \neq 1} V_{l\beta}(k_{l\beta}) \right] \prod_i A_i(\boldsymbol{k}_i) \cdot \sum_{x_j : \boldsymbol{k}_j} \iota(\kappa_{jm} = \alpha)\, \iota(\sigma_{jm} = 1)\, \nu_{j\alpha} \phi_j(m, k_j) \cdot f_j(k_j)\, b_{1\alpha}(\boldsymbol{k}_1). \tag{54}$$

We can reduce the sum in (54) to:

$$X_{j\alpha} = \sum_{\boldsymbol{k} \in S(\boldsymbol{K})} \prod_{\beta} \left[V_\beta(K_\beta - k_{1\beta} - \delta_{\alpha\beta}) \cdot \prod_{l \neq 1} V_{l\beta}(k_{l\beta} - \delta_{jl}\delta_{\alpha\beta}) \right] \cdot \prod_{i \neq j} A_i(\boldsymbol{k}_i)\, A_j(\boldsymbol{k}_j - \boldsymbol{u}_\alpha)\, p_{1j:\alpha}(\boldsymbol{k} - \boldsymbol{u}_{j\alpha}) \cdot b_{1\alpha}(\boldsymbol{k}_1)\, b_{j\alpha}(\boldsymbol{k}_j - \boldsymbol{u}_\alpha). \tag{55}$$

Here we used $\boldsymbol{u}_{j\alpha}$ to indicate that one job of class α was deleted from station j. In the V_β and $V_{l\beta}$ functions we have

used $\delta_{\alpha\beta}$ and δ_{jl} to indicate that a job of class α is missing from station j and the network. Doing this provides the factor $p_{1j;\alpha}(\boldsymbol{k} - \boldsymbol{u}_{j\alpha})$ that appears in (55). Equation (55) is the same as a sum over $S(\boldsymbol{K} - \boldsymbol{u}_\alpha)$, so we are justified in defining a set of functions:

$$H_{j\alpha}(\boldsymbol{K}) = \sum_{\boldsymbol{k}\in S(\boldsymbol{K})} \prod_{\beta} \left[V_\beta(K_\beta - k_{1\beta}) \prod_{l\neq 1} V_{l\beta}(k_{l\beta}) \right] \cdot \prod_i A_i(\boldsymbol{k}_i)\, p_{1j;\alpha}(\boldsymbol{k})\, b_{1\alpha}(\boldsymbol{k}_1)\, b_{j\alpha}(\boldsymbol{k}_j). \quad (56)$$

The sum in (55) is then $H_{j\alpha}(\boldsymbol{K} - \boldsymbol{u}_\alpha)$, and this is the claim for station j (49). The throughput of station 1 is simply the sum of the throughputs of the other stations, and adding equation (49) over all peripheral stations gives (50). □

V. Algorithm for the Computation of Performance Measures

We can express the functions G and $H_{i\alpha}$ as convolutions. With this, they can be computed by the convolution algorithm.

- $\boldsymbol{a}_1$ Array with elements $\Pi_\kappa V_\kappa(K_\kappa - k_{1\kappa})\, A_1(\boldsymbol{k}_1)$.
- $\boldsymbol{a}_j$ Array with elements $\Pi_\kappa V_{j\kappa}(k_{j\kappa})\, A_j(\boldsymbol{k}_j)$.
- $\boldsymbol{k}_{1\alpha}$ Array with elements $k_{1\alpha}\, \Pi_\kappa V_\kappa(K_\kappa - k_{1\kappa})\, A_1(\boldsymbol{k}_1)$.
- $\boldsymbol{k}_{j\alpha}$ Array with elements $k_{j\alpha}\, \Pi_\kappa V_{j\kappa}(k_{j\kappa})\, A_j(\boldsymbol{k}_j)$.
- $\boldsymbol{b}_{1\alpha}$ Array with elements $b_{1\alpha} w_\alpha(K_\alpha - k_{1\alpha}) \Pi_\kappa V_\kappa(K_\kappa - k_{1\kappa})\, A_1(\boldsymbol{k}_1)$.
- $\boldsymbol{b}_{j\alpha}$ Array with elements $b_{j\alpha} w_{j\alpha}(k_{j\alpha}) \Pi_\kappa V_{j\kappa}(k_{j\kappa})\, A_j(\boldsymbol{k}_j)$.

In what follows, we use * for convolution, $\div$ for deconvolution, and we take products of arrays as convolutions. The (array of) normalization constants can be written:

$$\boldsymbol{G} = \prod_i \boldsymbol{a}_i. \quad (57)$$

The (array of) mean number of jobs in station i is

$$\bar{\boldsymbol{k}}_{i\alpha} = (\boldsymbol{G} \div \boldsymbol{a}_i) * \boldsymbol{k}_{i\alpha}. \quad (58)$$

Define the auxiliary array $\boldsymbol{H}_{1\alpha}$ by

$$\boldsymbol{H}_{1\alpha} = (\boldsymbol{G} \div \boldsymbol{a}_1) * \boldsymbol{b}_{1\alpha}. \quad (59)$$

Then the (array of) $\boldsymbol{H}_{j\alpha}$ for $j \neq 1$ is given by

$$\boldsymbol{H}_{j\alpha} = (\boldsymbol{H}_{1\alpha} \div \boldsymbol{a}_j) * \boldsymbol{b}_{j\alpha}. \quad (60)$$

Equations (56)–(60) provide an efficient algorithm to compute G, $\bar{k}_{i\alpha}$, and the $H_{j\alpha}$. From these values the throughputs can be computed using Theorem 2. The mean sojourn times can then be obtained using Little's Law.

VI. Complexity of the Algorithm

Assume that all C job classes go to all peripheral stations. To compute G we do $(N - 1)$ convolutions. We need one deconvolution and one convolution to compute $H_{1\alpha}$. In this case we need to compute $C \cdot (N - 1)$ values of $H_{j\alpha}$, each of which needs a deconvolution and a convolution. This makes a total of $C \cdot (N - 1) + N + 1$ convolutions and deconvolutions. Each convolution or deconvolution involves $C \cdot K^C$ operations if there are K jobs in each class. So the time complexity of computing the throughputs is $O(C^2 N K^C)$. This contrasts with the convolution algorithm for classical (state independent and nonblocking) networks [23] where the total number of operations to compute the throughputs is $O(NCK^C)$. For the mean number of jobs again we have a time complexity of $O(C^2 N K^C)$. In case of classical networks the operation count for the mean number of jobs is the same. The space complexity is given by a fixed number of arrays, and so it is $O(K^C)$.

VII. Conclusions

We have shown that central server models with state-dependent routing and rejection blocking have product form equilibrium state probabilities. Using the exact equilibrium state distribution found, exact algorithms to compute performance measures are derived. The algorithms are more demanding than their counterparts for classical networks (nonblocking networks with state-independent routing). One of the most interesting properties of classical networks is that the distribution of states of the network as seen by a job arriving at a station is exactly the same as the equilibrium state distribution of the same network, only with the arriving job deleted. This is known as the Arrival Instant Distribution Theorem [24]. In view of the noted similarities it would be interesting to know whether there is a similar simple relation between the distributions at arrival instants at a station and the equilibrium state distribution. A related question is to look for relations between the distribution at the instants when a job is accepted and the equilibrium state distribution. In case of classical networks, the arrival instant distribution theorem is the basis for the mean value analysis, MVA [24]. It is doubtful that the corresponding arrival instant theorem and/or acceptance instant theorem (if they exist) for queueing networks with rejection blocking and state dependent routing can be used to construct a MVA-like algorithm.

References

[1] I. F. Akyildiz, "Exact analysis of queueing networks with rejection blocking," in *Proc. Int. Conf. Queueing Networks with Finite Capacities*. Amsterdam, The Netherlands: North-Holland, May 1988, pp. 21–33.

[2] I. F. Akyildiz and H. von Brand, "Exact solutions for open, closed and mixed queueing networks with rejection blocking," *Theoret. Comput. Sci.*, North-Holland, vol. 64, pp. 203–219, May 1989.

[3] ——, "Computational algorithms for networks of queues with rejection blocking," *Acta Inform.*, Springer-Verlag, to be published.

[4] ——, "Dual and selfdual networks of queues with rejection blocking," *Comput. J.*, Springer-Verlag, to be published.

[5] A. Barbour, "Networks of queues and the method of stages," *Adv. Appl. Prob.*, vol. 8, pp. 584–591, 1976.

[6] 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*, vol. 22, pp. 248–260, Apr. 1975.

[7] H. von Brand, "Queueing networks with blocking," Ph.D. dissertation, Dep. Comput. Sci., LSU, July 1987.

[8] J. P. Buzen, "Queueing network models of multiprogramming," Ph.D. dissertation, Div. Eng. Appl. Sci., Harvard Univ., Cambridge, MA, Aug. 1971.

[9] K. M. Chandy and A. J. Martin, "A characterization of product form queueing networks," *J. ACM*, vol. 30, pp. 286-299, 1983.
[10] K. M. Chandy, J. H. Howard, and D. F. Towsley, "Product form and local balance in queueing networks," *J. ACM*, vol. 24, no. 2, pp. 250-263, Apr. 1977.
[11] J. W. Cohen, "The multiple phase service network with generalized processor sharing," *Acta Inform.*, vol. 12, pp. 245-284, 1979.
[12] Y. Dallery and D. D. Yao, "Modeling a system of flexible manufacturing cells," in *Proc. Modeling and Design of Flexible Manufacturing Systems*. Amsterdam, The Netherlands: North-Holland, 1986, pp. 289-300.
[13] N. M. van Dijk and H. Tijms, "Insensitivity in two-node blocking models with applications," in *Proc. Teletraffic Analysis and Computer Performance Evaluation*. Amsterdam, The Netherlands: North-Holland, May 1986, pp. 329-340.
[14] N. M. van Dijk and I. F. Akyildiz, "Networks of mixed processor sharing parallel queues and common pools," Georgia Tech., Tech. Rep. GIT-ICS-88-022, June 1988.
[15] A. Hordijk and N. M. van Dijk, "Networks of queues with blocking," in *Proc. Performance '81*. Amsterdam, The Netherlands: North-Holland, 1981, pp. 51-65.
[16] ——, "Networks of queues. Part I: Job-local-balance and the adjoint process. Part II: General routing and service characteristics," in *Proc. Int. Conf. Modeling of Computer Systems*. New York: Springer-Verlag, Jan. 1983, vol. 60, pp. 158-205.
[17] F. P. Kelly, "Networks of queues with customers of different types," *J. Appl. Prob.*, vol. 12, pp. 542-554, 1975.
[18] ——, *Reversibility and Stochastic Networks*. New York: Wiley, 1979.
[19] A. E. Krzesinski, "Multiclass queueing networks with state-dependent routing," *Perform. Eval.*, vol. 7, no. 2, pp. 125-145, June 1987.
[20] A. S. Noetzel, "A generalized queueing discipline for product form network solutions," *J. ACM*, vol. 26, pp. 779-793, Oct. 1979.
[21] R. O. Onvural, "A survey on closed queueing networks with blocking," *ACM Comput. Surveys*, to be published.
[22] B. Pittel, "Closed exponential networks of queues with saturation. The Jackson-type stationary distribution and its asymptotic analysis," *Math. Oper. Res.*, pp. 357-378, 1979.
[23] M. Reiser and H. Kobayashi, "Queueing networks with multiple closed chains: Theory and computational algorithms," *IBM J. Res. Dev.*, vol. 19, pp. 283-294, 1975.
[24] M. Reiser and S. Lavenberg, "Mean value analysis of closed multichain queueing networks," *J. ACM*, vol. 27, no. 2, pp. 313-322, Apr. 1980; Corrigendum, vol. 28, no. 3, p. 629, 1981.
[25] C. H. Sauer, "Computational algorithms for state-dependent queueing networks," *ACM Trans. Comput. Syst.*, vol. 1, no. 1, pp. 67-92, Feb. 1983; Corrigendum, vol. 1, no. 4, p. 369, Nov. 1983.
[26] R. F. Serfozo, "Markovian network process: Congestion-dependent routing and processing," in *Queueing Systems: Theory and Applications (QUESTA)*, to be published.
[27] J. Spirn, "Queueing networks with random selection for service," *IEEE Trans. Software Eng.*, vol. SE-5, no. 3, pp. 287-289, May 1979.
[28] D. F. Towsley, "Queueing network models with state-dependent routing," *J. ACM*, vol. 27, no. 2, pp. 323-337, Apr. 1980.
[29] D. D. Yao and J. A. Buzacott, "Modeling a class of state-dependent routing in flexible manufacturing systems," *Ann. Oper. Res.*, vol. 3, pp. 153-167, 1985.

I. F. Akyildiz (M'85-SM'89) was born in 1954 in Istanbul, Turkey. He received the Vordiplom, Diplom Informatiker, and Doctor of Engineering degrees in computer science from the University of Erlangen—Nuernberg, West Germany, in 1978, 1981, and 1984, respectively.

Currently, he is an Associate Professor in the School of Information and Computer Science at Georgia Institute of Technology, Atlanta. He is a coauthor of a textbook entitled *Analysis of Computer Systems* published by Teubner Verlag in the Fall of 1982. He is a guest editor of the special issue on *queueing networks with finite capacities* in *Performance Evaluation Journal*. He is also an associate editor for *Computer Networks* and *ISDN Journal*. His research interests are performance evaluation, computer networks, distributed systems, and computer security.

Dr. Akyildiz is a member of ACM (Sigops and Sigmetrics), and GI (Gesellschaft fuer Informatik).

Horst von Brand was born on August 22, 1954 in Valparaiso, Chile. He got the Ingeniero Civil Quimico degree from the Universidad Tecnica Federico Santa Maria, Valparaiso, in 1981, and the Ph.D. degree in computer science from Louisiana State University in 1987.

Since 1982, he has been with the Departmento de Informatica of the Universidad Tecnica Federico Santa Maria. His current research interests are in performance modeling, object oriented programming, and expert systems.