Dynamic Scaling and Growth Behavior of Queuing Network Normalization Constants

SIMON S. LAM

University of Texas at Austin, Austin, Texas

ABSTRACT. A simple dynamic scaling technique is shown that avoids both the overflow and underflow problems that are often encountered in the evaluation of normalization constants of closed product-form queuing networks With dynamic scaling, normalization constants for very large routing chain population sizes can be evaluated within the bounds of a relatively small range of numbers. It is shown that the product-form solution possesses a local balance property and the $M \Rightarrow M$ property with respect to routing chains. The relationships between normalization constants of closed networks and certain equilibrium aggregate state probabilities in networks that permit external arrivals and departures are examined. The growth behavior of normalization constants is shown to be modeled by a birth–death process traversing over the set of chain population vectors

Categories and Subject Descriptors: C 2 4 [**Computer-Communication Networks**]: Distributed Systems—*network operating systems*; D.4.4 [**Operating Systems**]: Communications Management—*network communication*; D.4.8 [**Operating Systems**]: Performance—*modeling and prediction; queuing theory*

General Terms: Performance, Theory

Additional Key Words and Phrases: Queuing networks, product-form solution, normalization constants, dynamic scaling, local balance, Poisson departures, population size constraints

1. *Introduction*

Queuing networks have been used extensively and successfully in the modeling of computer systems and communication networks. Jackson [7] first showed that the equilibrium probability distribution $P(S)$ of the state S of a network of first-come-first-served queues is in the form of a product of terms that correspond to the state probabilities of the individual queues considered in isolation. Presently, most known networks with an exact solution for $P(S)$ belong to the class of BCMP networks discovered and characterized by Baskett, Chandy, Muntz, and Palacios [1, 3, 11]. Four types of service centers as well as open and closed routing chains are allowed.

BCMP networks have a product-form solution for $P(S)$. This product-form solution was later shown to be applicable also to an extended class of BCMP networks with constraints on chain population sizes [8].

The product-form solution needs to be divided by a normalization constant to form a proper probability distribution for $P(S)$. The normalization constant is simply the sum of the product-form solution over all feasible network states. Since the number of feasible network states is typically very large, the summation is a nontrivial process.

This work was supported by the National Science Foundation under Grant ECS 78-01803. This work originally appeared as a technical report entitled, "Behavior of the Normalization Constant and a Scaling Technique for Product-Form Queueing Networks" [9]

Author's address. Department of Computer Sciences, University of Texas, Austin, TX 78712.

Several computational algorithms are available for the class of BCMP networks [2, 5, 10, 14, 15]. The convolution algorithm was first discovered by Buzen [2] for single-chain networks and extended by Reiser and Kobayashi [14] to multichain networks. The LBANC and CCNC algorithms were recently proposed by Chandy and Sauer [5]. These algorithms all attempt first to evaluate the normalization constants of networks of closed chains. Network performance measures are then computed from the normalization constants. A major difficulty often encountered in the evaluation of the normalization constant $G(\mathbf{N})$ of a network with population vector $\mathbf{N}$ using any of these algorithms is that as the chain population sizes in $\mathbf{N}$ become large, $G(\mathbf{N})$ may become too large (causing a floating-point overflow) or too small (causing a floating-point underflow) [5, 13]. A scaling technique was described by Reiser [13] that can avoid the overflow problem. However, the bound used is not very tight, and no solution is provided for the underflow problem. The mean-value-analysis (MVA) algorithm proposed by Reiser and Lavenberg [15] bypasses the evaluation of $G(\mathbf{N})$ and computes various network performance measures directly.

SUMMARY OF OUR RESULTS. The overflow and underflow problems encountered in the evaluation of $G(\mathbf{N})$ using current algorithm implementations result from the use of a fixed set of "scaling factors" for the entire range of values of $\mathbf{N}$ of interest. We found that the scaling factors can be factored out of the expression for $G(\mathbf{N})$ so that one can easily use different sets of scaling factors for different values of $\mathbf{N}$ with just small amounts of space and computation overheads. As a result, the scaling factors can be changed to smaller values when $G(\mathbf{N})$ is about to encounter an overflow and to larger values when $G(\mathbf{N})$ is about to encounter an underflow. Since changes in the values of scaling factors can be made repeatedly during the execution of a computational algorithm, it is now possible to evaluate $G(\mathbf{N})$ for a wide range of values of $\mathbf{N}$ using a small range of floating-point numbers or even fixed-point numbers! The scaling technique and related results are covered in Section 3.

External Poisson arrivals at rates that may depend upon routing chain population sizes are allowed in BCMP networks [1] and the extended class of BCMP networks with population size constraints [8]. In such a network the population vector $\mathbf{N}$ changes as a result of external arrivals into the network or customer departures from the network. We have shown that class local balance [1, 3] implies chain local balance. Furthermore, routing chains possess the $M \Rightarrow M$ property [11]. The equilibrium probability of the aggregate of feasible network states with population vector $\mathbf{N}$ is related to the normalization constant of a closed network with the same population vector. These equilibrium probabilities are equal to the equilibrium state probabilities of a birth–death process traversing over the set of population vectors. The growth behavior of normalization constants is thus modeled by such a birth–death process with birth rates equal to scaling factors and state-dependent death rates. These results are covered in Section 4.

2. *Definitions and Notation*

Service centers are indexed by $m = 1, 2, \ldots, M$. Customers belong to different chains with different routing behaviors and service requirements. Chains are indexed by $k = 1, 2, \ldots, K$. Let there be C classes in the network. At any time each customer must be in one of the C classes but may make a transition to another class some time later. Classes are used to model a customer's routing behavior and service requirements with finite memory.

The set of classes $\{1, 2, \ldots, C\}$ is partitioned in two different ways. First, they are partitioned over the set of M service centers. We let $\mathrm{SC}(m)$ denote the partition of

classes belonging to service center m. Thus the class of a customer, say, c in SC(m), uniquely identifies the service center he is in. A customer makes a transition from class c to class d with probability p_{cd}. The transition from class c to class d may correspond to a transition of the customer from one service center to another if c and d belong to different service centers, or it may correspond to a transition of the customer from one class to another within the same center.

The set of classes $\{1, 2, \ldots, C\}$ is also partitioned over the set of K chains. We let RC(k) denote the partition of classes belonging to routing chain k. Customers cannot make transitions between classes belonging to two different chains. (Otherwise, the two different chains "communicate" and should be treated as just one chain.) In other words, $p_{cd} = 0$ if c and d are in different chains. Moreover, each chain is irreducible, that is, the transition probabilities $\{p_{cd}, c, d \text{ in } \mathrm{RC}(k)\}$ are such that every class can reach every other class in the same chain in a finite number of transitions with nonzero probability.

For each chain $k = 1, 2, \ldots, K$, the relative arrival rates of customers to the different classes can be determined (to within a multiplicative constant) by solving the set of equations

$$v_d = \sum_{c \text{ in } \mathrm{RC}(k)} v_c p_{cd}, \qquad d \text{ in } \mathrm{RC}(k). \tag{1}$$

Summing over the different classes in a service center, the relative arrival rate of chain-k customers to center m is

$$\lambda_{mk} = \sum_{\substack{c \text{ in } \mathrm{SC}(m) \\ \text{and } \mathrm{RC}(k)}} v_c. \tag{2}$$

Suppose that the multiplicative constant in (1) is chosen such that

$$\lambda_{1k} = \alpha_k.$$

For $\alpha_k = 1$, λ_{mk} is equal to the mean number of visits to center m by a chain-k customer between successive visits to center 1. α_k is called the *scaling factor* of chain k. (Note that since the labeling of the service centers is arbitrarily done, the choice of center 1 is arbitrary.)

Let τ_c denote the mean service time of a customer in class c (assuming that he is served at the rate of 1 second of work required per second). The mean service time of chain-k customers at center m is

$$\tau_{mk} = \sum_{\substack{c \text{ in } \mathrm{SC}(m) \\ \text{and } \mathrm{RC}(k)}} \frac{v_c}{\lambda_{mk}} \tau_c. \tag{3}$$

The traffic intensity of chain-k customers through center m is defined to be

$$\rho_{mk} = \lambda_{mk}\tau_{mk} = \sum_{\substack{c \text{ in } \mathrm{SC}(m) \\ \text{and } \mathrm{RC}(k)}} v_c \tau_c. \tag{4}$$

We define the *nominal traffic intensity* to be

$$w_{mk} = \lambda_{mk}\tau_{mk} \qquad \text{for} \quad \alpha_k = 1. \tag{5}$$

Thus we have

$$\rho_{mk} = \alpha_k w_{mk}. \tag{6}$$

The service rate of a service center may depend upon the number of customers currently in the center. Let $\mu_m(i)$ denote the service rate of center m containing i customers. A service center is said to be *fixed-rate* if $\mu_m(i) = 1$.

For the moment we consider only networks with closed chains. (Networks that permit departures and external arrivals are introduced in Section 4.) We let N_k be the number of customers in chain k. The network *population vector* is

$$\mathbf{N} = (N_1, N_2, \ldots, N_K).$$

The normalization constant for a closed network with population vector $\mathbf{N}$ is denoted $G(\mathbf{N})$.

Let n_{mk} denote the number of chain-k customers in center m. Define the network state

$$\mathbf{n} = (\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_M),$$

where

$$\mathbf{n}_m = (n_{m1}, n_{m2}, \ldots, n_{mK}), \qquad m = 1, 2, \ldots, M.$$

(We note that $\mathbf{n}$ is non-Markovian and corresponds to an aggregation of detailed network states that are Markovian.) The product-form solution for a BCMP closed network with population vector $\mathbf{N}$ [1] is

$$P(\mathbf{n}) = \frac{\prod_{m=1}^{M} p_m(\mathbf{n}_m)}{G(\mathbf{N})}, \tag{7}$$

where

$$p_m(\mathbf{n}_m) = \left\{ \prod_{i=1}^{n_m} \frac{1}{\mu_m(i)} \right\} n_m! \prod_{k=1}^{K} \frac{\rho_{mk}^{n_{mk}}}{n_{mk}!}, \tag{8}$$

where

$$n_m = n_{m1} + n_{m2} + \cdots + n_{mK}.$$

The form of eq. (8) is the same for all four types of service centers considered in [1]; they are: first-come-first-served (FCFS), processor-sharing (PS), last-come-first-served preemptive resume (LCFSPR), and infinite servers (IS). However, in an FCFS center it is necessary for the mean service time to be independent of class membership, that is, $\tau_c = \tau_m$ for any c in SC(m). Also, an IS center, say m, assumes that $\mu_m(i) = i$ for all feasible i.

Finally, the normalization constant is by definition

$$G(\mathbf{N}) = \sum_{\substack{\mathbf{n} \text{ such that} \\ \sum_{m=1}^{M} \mathbf{n}_m = \mathbf{N}}} \prod_{m=1}^{M} p_m(\mathbf{n}_m). \tag{9}$$

In addition to service-rate functions of the form $\mu_m(i)$ described above, two other forms of state-dependent service rates are allowed in BCMP networks [1]. The second form of state-dependent service rates distinguishes customers belonging to different classes. The service rate of customers belonging to a specific class may be a function of the number of customers in that class (this form does not apply to classes within a FCFS service center). The third form of state-dependent service rates involves the total number of customers in a set, say I, of service centers. The service rates of customers in different service centers in I may be functions of the total number of customers in those centers, that is, $\sum_{m \in I} n_m$. To accommodate these two other forms of state-dependent service rates, the product-form solution needs to be generalized

slightly. (Hence, eqs. (7)–(9) above, as well as eqs. (22), (25), and (27) below need to be generalized slightly; see [1].).

To keep the notation and equations simple in this paper, we shall not explicitly consider these other forms of service-rate functions. It is, however, easy to show that the new results and observations presented in this paper are applicable to networks with any or all of the three forms of state-dependent service rates.

3. *Growth Behavior and Dynamic Scaling of Normalization Constants*

Examining eqs. (8) and (9), we note that $G(\mathbf{N})$ is a function of $\mathbf{N}$, M, the service rate functions $\{\mu_m(i)\}$, and the traffic intensities $\{\rho_{mk}\}$. Recall that ρ_{mk} is the product of the scaling factor α_k and the nominal traffic intensity w_{mk}. Let

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K).$$

In what follows we shall often use the notation $G(\boldsymbol{\alpha}, \mathbf{N})$ or $G(\boldsymbol{\alpha}, M, \mathbf{N})$ instead of $G(\mathbf{N})$ to explicitly indicate the parameters $\boldsymbol{\alpha}$ and M assumed in the normalization constant. Our scaling technique, to be described later, makes use of the following lemma.

LEMMA 1

$$G(\boldsymbol{\alpha}, M, \mathbf{N}) = \alpha_1^{N_1}\alpha_2^{N_2} \cdots \alpha_K^{N_K} G(\mathbf{1}, M, \mathbf{N}), \tag{10}$$

where $\mathbf{1}$ is a K-vector of ones denoting that the scaling factor is equal to unity for each chain.

A useful corollary of the above lemma is

$$G(\boldsymbol{\beta}, M, \mathbf{N}) = r(\boldsymbol{\beta}, \boldsymbol{\alpha}, \mathbf{N}) G(\boldsymbol{\alpha}, M, \mathbf{N}), \tag{11}$$

where

$$r(\boldsymbol{\beta}, \boldsymbol{\alpha}, \mathbf{N}) = \prod_{k=1}^{K} \left(\frac{\beta_k}{\alpha_k}\right)^{N_k}$$

The above lemma is obvious from a careful inspection of the definition of $G(\mathbf{N})$ in eq. (9) and noting that the summation is over those values of $\mathbf{n}$ such that $\sum_{m=1}^{M} \mathbf{n}_m = \mathbf{N}$.

It is instructive, however, to demonstrate the above lemma by a different approach. It is well known that the throughput rate of chain-k customers at center m for a network with population vector $\mathbf{N}$ [2, 5, 14] is given by

$$T_{mk}(\mathbf{N}) = \lambda_{mk} \frac{G(\mathbf{N} - \mathbf{1}_k)}{G(\mathbf{N})} \qquad \text{for any} \quad m \quad \text{and} \quad \mathbf{N} \geq \mathbf{1}_k, \tag{12}$$

where $\mathbf{1}_k$ is a K-vector with the kth element equal to one and all others equal to zero. The relation $\geq$ between two vectors is satisfied if it is satisfied for each pair of corresponding components in the vectors. Equation (12) can be rewritten as

$$G(\mathbf{N}) = \frac{\lambda_{mk}}{T_{mk}(\mathbf{N})} G(\mathbf{N} - \mathbf{1}_k) \qquad \text{for any} \quad m \quad \text{and} \quad \mathbf{N} \geq \mathbf{1}_k.$$

A consequence of eq. (12) is that the ratio $\lambda_{mk}/T_{mk}(\mathbf{N})$ is constant over m. Let us consider $m = 1$. Recall that λ_{1k} is equal to the scaling factor α_k by definition. To simplify our notation, we shall write $T_k(\mathbf{N})$ for $T_{1k}(\mathbf{N})$. The above equation can now

be rewritten as

$$G(\mathbf{N}) = \frac{\alpha_k}{T_k(\mathbf{N})} G(\mathbf{N} - \mathbf{1}_k), \qquad \mathbf{N} \geq \mathbf{1}_k. \tag{13}$$

Traditionally, we first compute $G(\mathbf{N})$ and then derive $T_k(\mathbf{N})$ from $G(\mathbf{N})$ and $G(\mathbf{N} - \mathbf{1}_k)$. Now since we are interested in the behavior of $G(\mathbf{N})$, we consider the reverse process. Note that $T_k(\mathbf{N})$ can be obtained from the MVA algorithm directly and is independent of the scaling factor α_k [15].

We need some additional notation at this point. Consider, in the K-dimensional space of population vectors, a path leading from the vector $\mathbf{0}$ of all zeros to $\mathbf{N}$. The path has

$$N = N_1 + N_2 + \cdots + N_K$$

steps. Step i in the path corresponds to the addition of a chain-k_i customer to the current population vector $\mathbf{N}^{(i-1)}$. The increasing sequence of population vectors along the path is

$$\begin{aligned}
\mathbf{N}^{(0)} &= \mathbf{0} \\
\mathbf{N}^{(1)} &= \mathbf{N}^{(0)} + \mathbf{1}_{k_1}, \\
\mathbf{N}^{(2)} &= \mathbf{N}^{(1)} + \mathbf{1}_{k_2}, \\
&\vdots \\
\mathbf{N}^{(N)} &= \mathbf{N}^{(N-1)} + \mathbf{1}_{k_N} = \mathbf{N}.
\end{aligned}$$

Given any such path, a solution for $G(\mathbf{N})$ using the recursive relation in eq. (13) is

$$G(\mathbf{N}) = \frac{\alpha_1^{N_1}\alpha_2^{N_2} \cdots \alpha_K^{N_K}}{\prod_{i=1}^{N} T_{k_i}(\mathbf{N}^{(i)})}, \tag{14}$$

where $G(\mathbf{0}) = 1$ by definition. We have thus provided an alternate proof of Lemma 1.

Note that there are many different paths leading from $\mathbf{0}$ to $\mathbf{N}$. Since $G(\mathbf{N})$ is a constant, the next lemma is immediately obvious.

LEMMA 2. *For any path from* $\mathbf{0}$ *to* $\mathbf{N}$ *consisting of an increasing sequence of population vectors* $\mathbf{N}^{(1)}, \mathbf{N}^{(2)}, \ldots, \mathbf{N}^{(N-1)}, \mathbf{N}^{(N)}$,

$$\prod_{i=1}^{N} T_{k_i}(\mathbf{N}^{(i)}) = \textit{constant}. \tag{15}$$

Let us set aside the above result until Section 4. We shall now consider the special case of $K = 1$, that is, networks with a single chain, and introduce a dynamic scaling technique for avoiding the overflow/underflow problems. The scaling technique for networks with multiple chains is similar and will be considered afterward.

For a network with a single closed chain our previous notation is simplified as follows:

$G(N)$	normalization constant for N customers in the chain;
α	scaling factor (relative arrival rate at center 1);
$T(N)$	throughput rate at center 1 for N customers in the chain.

We now have

$$G(N) = \frac{\alpha}{T(N)} G(N-1), \qquad N \geq 1,$$

and with $G(0) = 1$ by definition, we have

$$G(N) = \alpha^N \prod_{i=1}^{N} \frac{1}{T(i)}. \tag{16}$$

To characterize the behavior of $T(i)$, we assume for the moment that service-rate functions are limited to

$$\mu_m(i) = \begin{cases} i, & 1 \le i \le j_m, \\ j_m, & i \ge j_m, \end{cases} \tag{17}$$

for any m, and state the following result.

PROPOSITION. *$T(N)$ is monotonically nondecreasing in N.*

The above proposition was proved by Chang and Lavenberg [6] for a network of FCFS centers. Their proof is also valid for IS centers, since j_m can be greater than N. Moreover, we note that any BCMP single-chain network with the same set of service-center traffic intensities $\{\rho_m\}$ has the same marginal probability distributions $P_m(n_m)$, $m = 1, 2, \ldots, M$, which together with $\mu_m(i)$ determine the service-center throughput rates. Consequently, the above proposition applies to any product-form network with a single chain and the service-rate functions of eqs. (17).

We can also calculate the limiting value of $T(N)$ as $N \to \infty$. Recall that w_m denotes the nominal traffic intensity of center m. The relative utilization of center m is defined to be

$$u_m = \frac{w_m}{j_m},$$

where j_m is the maximum service rate of center m. Let m^* denote the service center with the largest relative utilization, that is,

$$u_{m^*} = \max_m u_m.$$

As $N \to \infty$, center m^* becomes the bottleneck in the network with an infinite queue and an actual utilization of unity [12]. The limiting throughput of center m is thus

$$\lim_{N\to\infty} T_m(N) = \frac{u_m}{u_{m^*}} \frac{j_m}{\tau_m},$$

in customers served per second. Specifically, we have for center 1

$$T(N) \le \frac{u_1}{u_{m^*}} \frac{j_1}{\tau_1} \triangleq T_{\max}. \tag{18}$$

The typical behavior of $T(N)$ as a function of N is plotted in Figure 1.

Referring back to eq. (16), we can now show that the behavior of the normalization constant $G(N)$ depends upon the relative magnitudes of the scaling factor α and $T_{\max}$. The three general cases of behavior are illustrated in Figure 2. We see that if $\alpha \ge T_{\max}$, we can potentially have an overflow problem due to $G(N)$ getting very large. If $\alpha < T_{\max}$ and as N increases, we can potentially first encounter an overflow as $G(N)$ increases and then an underflow problem as $G(N)$ subsequently decreases.

EXAMPLES ILLUSTRATING DYNAMIC SCALING. Current computational algorithms assume the use of the same scaling factor α to compute $G(N)$ for the full range of N values of interest. Lemma 1 and eq. (11) show that the scaling factor can be easily changed at any time during the computational process. We only need to remember what values of α were used for specific values of N. To illustrate such a dynamic

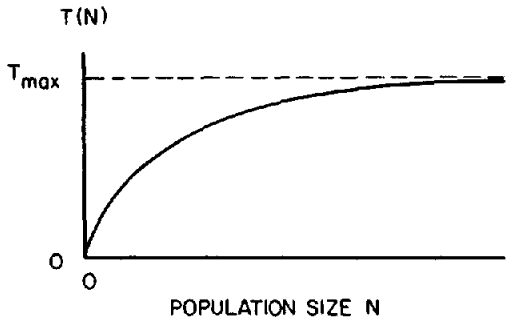

FIG 1. Throughput rate versus population size in a single-chain network.

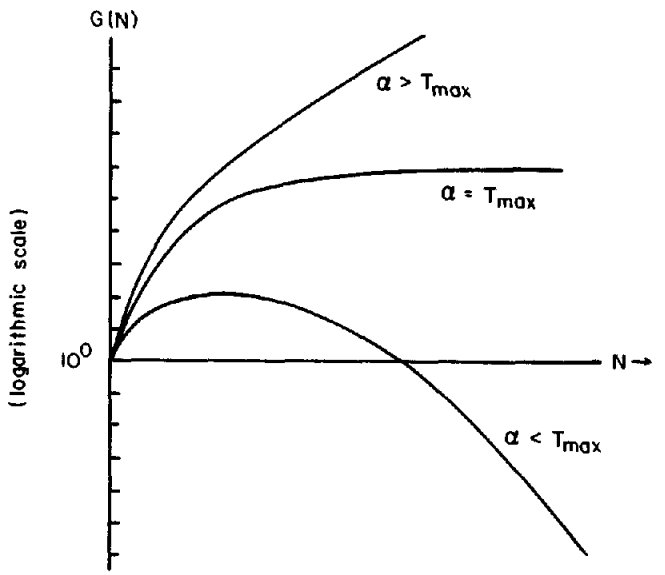

FIG. 2. Behavior of $G(N)$ in a single-chain network.

scaling technique, we use an example considered by Chandy and Sauer in [5] and illustrated in Figure 3. Center 4 is an IS center that models a population of terminals. Centers 1–3 are all fixed-rate centers. The relative arrival rates λ_m (at $\alpha = 1$) and mean service times τ_m are as follows:

m	λ_m	τ_m
1	1	0.020
2	0.2	0.044
3	0.8	0.008
4	0.2	15

In Figure 4, $G(N)$ is shown as a function of N for different values of α. Suppose we need to compute $G(100)$ on a computer that can only represent floating numbers between 10^{-10} to 10^{10}. A dynamic scaling approach then is to start with an arbitrary scaling factor, say $\alpha = 50$, as shown in Figure 4. When a floating-point overflow is about to occur, α is changed to a smaller value using eq. (11). When a floating-point underflow is about to occur, α is changed to a larger value. As shown in Figure 4, after several changes in α we finally found $G(100) = 0.1430$ for $\alpha = 12.5$ without exceeding the 10^{-10}–10^{10} floating-point range. It was not unlikely that we ended up with a scaling factor that we used earlier, but the scaling technique enabled us to

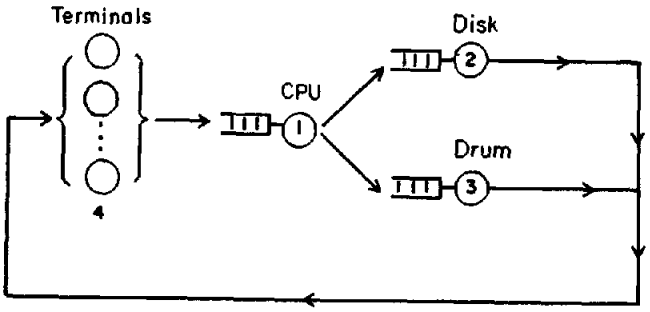

FIG. 3. Single-chain network example

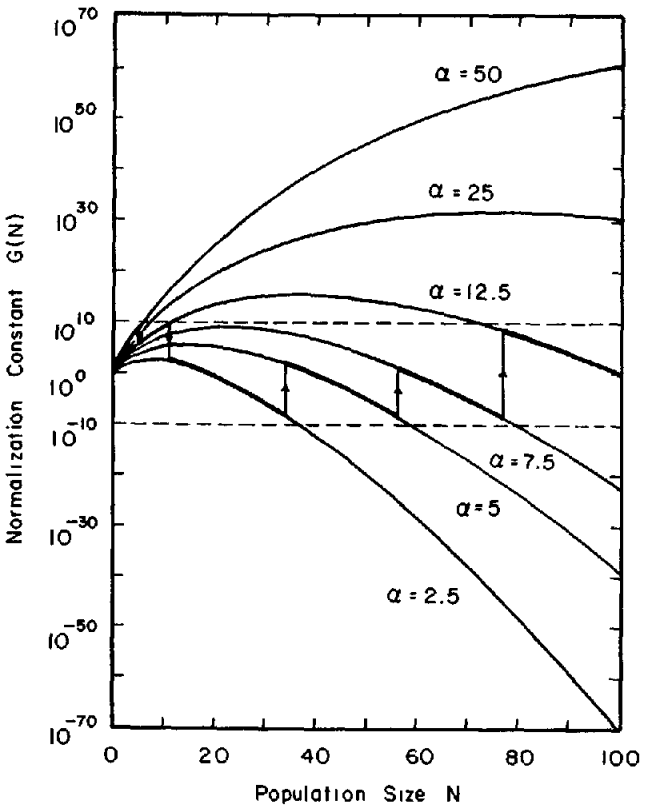

FIG. 4. Dynamic scaling for single-chain network example.

bypass the interval of N values within which we cannot represent $G(N)$ using that scaling factor.

We next consider networks with more than one routing chain. In this case the above proposition no longer applies. We note, however, that the monotone property in the proposition is not necessary for doing dynamic scaling.

Consider the following example of a network of three fixed-rate centers with two routing chains. The nominal traffic intensities w_{nk} (for $\alpha_1 = \alpha_2 = 1$) are

	center 1	center 2	center 3
chain 1	2	4	2
chain 2	2	4	1

TABLE I. NORMALIZATION CONSTANTS AND THEIR SCALING FACTORS FOR THE TWO-CHAIN NETWORK EXAMPLE

		(N_1, N_2)								
		(0, 0)	(0, 1)	(1, 0)	(1, 1)	(0, 2)	(2, 0)	(1, 2)	(2, 1)	(2, 2)
m = 1	G	1	2	2	8	4	4	24	24	64
	α	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)
m = 2	G	1	6	6	56	28	28	30	20	$30\frac{4}{9}$
	α	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	$\left(\frac{1}{3}, \frac{1}{2}\right)$	$\left(\frac{1}{3}, \frac{1}{2}\right)$	$\left(\frac{1}{6}, \frac{1}{2}\right)$
m = 3	G	1	7	8	78	35	44	$21\frac{1}{6}$	$7\frac{7}{9}$	$41\frac{7}{18}$
	α	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)	$\left(\frac{1}{6}, \frac{1}{2}\right)$	$\left(\frac{1}{6}, \frac{1}{2}\right)$	$\left(\frac{1}{6}, \frac{1}{2}\right)$

Let us employ the convolution algorithm for fixed-rate servers from [14]. Let $G(\alpha, m, \mathbf{N})$ denote the normalization constant for the *first m centers* with scaling factors α and population vector $\mathbf{N}$. We have

$$G(\alpha, m, \mathbf{N}) = G(\alpha, m-1, \mathbf{N}) + \sum_{k=1}^{K} G(\alpha, m, \mathbf{N} - \mathbf{1}_k)\rho_{mk} \qquad \text{for} \quad m \geq 2,$$

and

$$G(\alpha, 1, \mathbf{N}) = n_1! \prod_{k=1}^{K} \frac{\rho_{1k}^{n_{1k}}}{n_{1k}!}.$$

The above recursive equation can be rewritten as

$$G(\alpha, m, \mathbf{N}) = r(\alpha, \beta, \mathbf{N})G(\beta, m-1, \mathbf{N}) + \sum_{k=1}^{K} r(\alpha, \gamma, \mathbf{N} - \mathbf{1}_k)G(\gamma, m, \mathbf{N} - \mathbf{1}_k)\alpha_k w_{mk}, \tag{19}$$

where $r(\alpha, \beta, \mathbf{N})$ was defined earlier. Suppose in the two-chain network example we want the normalization constant for $\mathbf{N} = (2, 2)$. However, the largest value of the normalization constant that we can store is 100. By dynamically changing the scaling factors and employing eq. (19) we arrived at the results tabulated in Table I.

COMPUTATION OF PERFORMANCE MEASURES. As illustrated in the above example, when the normalization constants of more than one population vector are used in the same formula, they need to have the same scaling factors.

Service-center throughput rates can be computed using the formula

$$T_{mk}(\mathbf{N}) = \lambda_{mk} \frac{G(\alpha, M, \mathbf{N} - \mathbf{1}_k)}{r(\alpha, \beta, \mathbf{N})G(\beta, M, \mathbf{N})}, \tag{20}$$

where it is assumed that $\lambda_{1k} = \alpha_k$. The mean queue size $q_{mk}(\mathbf{N})$ for a fixed-rate service center can be computed using the formula

$$q_{mk}(\mathbf{N}) = \alpha_k w_{mk} \frac{G_{m+}(\alpha, M, \mathbf{N} - \mathbf{1}_k)}{r(\alpha, \beta, \mathbf{N})G(\beta, M, \mathbf{N})}, \tag{21}$$

where G_{m+} is the output of the convolution algorithm over centers $1 - M$ but with center m convolved twice [14]. In both cases, since the normalization constants needed range over population vectors that differ by one customer, finding a set of scaling factors to fit the normalization constants within a given floating-point range should not pose much of a problem.

A difficulty may arise in the calculation of the mean queue length for a service center for which $\mu_m(i)$ is not a constant. In this case the marginal queue-length distribution may need to be first computed as follows:

$$P_m(\mathbf{n}_m) = \frac{p_m(\mathbf{n}_m)G_{m-}(\alpha, M, \mathbf{N} - \mathbf{n}_m)}{r(\alpha, \beta, \mathbf{N})G(\beta, M, \mathbf{N})}, \tag{22}$$

where $p_m(\mathbf{n}_m)$ was defined earlier. G_{m-} is the output of the convolution algorithm over centers $1 - M$ but skipping over center m. Since $\mathbf{n}_m$ may range from $\mathbf{0}$ to $\mathbf{N}$, it will then be likely that we cannot fit the normalization constants of $\mathbf{N} - \mathbf{n}_m$ and $\mathbf{N}$ within a given floating-point range using the same scaling factors. However, we observe that if the floating-point range is of reasonable size, then the mean queue length can still be computed accurately by simply discarding those marginal queue-length probabilities $P_m(\mathbf{n}_m)$ that are too small and will cause underflows! Let SMALLEST (LARGEST) denote the smallest (largest) floating-point number available. The error introduced in the mean queue length is negligible if

$$\left(\sum_{n=1}^{N_k} n\right)\text{SMALLEST} \ll \text{LARGEST} \qquad \text{for any} \quad k.$$

The above will hold given any nontrivial floating-point range and reasonable chain population sizes.

Space Overhead Considerations. The additional space overhead of dynamic scaling depends upon the computational algorithm and its implementation. In a convolution algorithm the recursion is done over the service centers. Consequently, an entire array of normalization constants for all population vectors between $\mathbf{0}$ and $\mathbf{N}$ is needed. A straightforward way to provide a mapping between population vectors and their corresponding scaling factors is to provide an entire array of α values. However, an inspection of the example in Table I suggests that since changes occur infrequently, it is possible to provide the mapping between population vectors and scaling factors with substantially less memory than that of an entire array. In the LBANC algorithm the recursion is done over the population vectors; hence additional saving is possible, since an entire array, indexed from $\mathbf{0}$ to $\mathbf{N}$, of normalization constants is not needed.

The amount of space overhead of dynamic scaling for any computational algorithm can be reduced significantly with the use of the same scaling factor, say α, for all chains (at the expense of, perhaps, some flexibility). This way, only the mapping between N $(= N_1 + N_2 + \cdots + N_K)$ and α needs to be remembered and can be accomplished with a minimal amount of space overhead; specifically, only the values of N at which a scaling change occurs need be remembered. In this case let $G(\alpha, M, \mathbf{N})$ be the normalization constant that we want to scale down (or up). Scaling can be simply accomplished by updating the pair of values of G and α for the given M and $\mathbf{N}$ as follows:

$$\alpha \leftarrow \beta\alpha, \qquad G \leftarrow \beta^N G,$$

where $N = N_1 + N_2 + \cdots + N_K$. Let LARGE and SMALL be floating numbers such that

$$\text{SMALLEST} < \text{SMALL} < 1 < \text{LARGE} < \text{LARGEST}.$$

Suppose we want to keep G within the range (SMALL, LARGE). When G exceeds LARGE, it can be scaled down to near unity with the choice of

$$\beta \leftarrow \frac{1}{(\text{LARGE})^{1/N}}.$$

When G drops below SMALL, it can be scaled up to near unity with the choice of

$$\beta \leftarrow \frac{1}{(\text{SMALL})^{1/N}}.$$

TIME OVERHEAD CONSIDERATIONS. The additional time overhead of dynamic scaling depends upon its implementation which, in turn, depends upon the computational algorithm and the programming language involved.

The implementation of dynamic scaling is simplest if the programming language has provisions for detecting underflows/overflows and recovering from them. In this case the additional time overhead of dynamic scaling is rather insignificant. Each time the scaling factors are changed, eq. (11) needs to be computed. Assuming that the available floating-point range is not too small and $G(\mathbf{N})$ does not fluctuate greatly as a function of $\mathbf{N}$ (owing to fluctuations in $\mu_m(i)$), the frequency of encountering overflow or underflow conditions requiring a change in scaling factors should be very low.

If overflows/underflows cannot be easily detected and recovered from, then it is necessary to prevent their occurrence by testing the magnitude of operands before operations. Upper and lower bounds, LARGE and SMALL, respectively, are needed. Rescaling is required if not all operands lie within these bounds. The basic trade-off in the design of an implementation algorithm is then as follows. In the extreme case, if operands are tested before every arithmetic operation, then the range (SMALL, LARGE) can be chosen to be close to the floating-point range of (SMALLEST, LARGEST). However, given network parameters, bounds may be obtained for SMALL and LARGE so that the testing of operands needs to be performed only once for each group of operations (e.g., subroutine for one convolution, subroutine for mean-queue-length computation, etc.).

4. *A General Queuing Network Model*

The queuing network model described in Section 2 is for closed routing chains, each with a fixed number of circulating customers. The model will now be extended to include chains that can have external arrivals and departures. External customer arrival streams to the chains are assumed to be Poisson processes. It is also assumed that a new external arrival to chain k joins class c with probability q_c, so that

$$\sum_{c \text{ in } \mathrm{RC}(k)} q_c = 1.$$

To determine the set of arrival rates $\{\lambda_{mk}\}$ for use in the traffic intensities $\{\rho_{mk}\}$, the following set of equations should be used (instead of eq. (1)):

$$v_d = q_d + \sum_{c \text{ in } \mathrm{RC}(k)} v_c p_{cd}, \qquad d \text{ in } \mathrm{RC}(k), \tag{23}$$

$$\lambda_{mk} = \sum_{\substack{c \text{ in } \mathrm{SC}(m) \\ \text{and } \mathrm{RC}(k)}} v_c. \tag{24}$$

There can be two types of Poisson arrival processes.

Type 1. The arrival rate of chain-k customers is a function of the total network population N, $\gamma_k(N)$, $k = 1, 2, \ldots, K$. Define

$$\gamma(N) = \gamma_1(N) + \gamma_2(N) + \cdots + \gamma_K(N).$$

Type 2. The arrival rate of chain-k customers is a function of the number of chain-k customers in the network, $\gamma_k(N_k)$, $k = 1, 2, \ldots, K$.

For networks with K chains, each of which may be open or closed, Baskett et al. [1] showed that the product-form solution in eq. (7) becomes

$$P(\mathbf{n}) = \frac{a(\mathbf{n})}{G} \prod_{m=1}^{M} p_m(\mathbf{n}_m), \tag{25}$$

where $p_m(\mathbf{n}_m)$ was given by eq. (8), G is the normalization constant and is equal to the sum of the unnormalized solution in eq. (25) over all feasible $\mathbf{n}$ states, and

$$a(\mathbf{n}) = \begin{cases} \prod_{i=0}^{N(\mathbf{n})-1} \gamma(i) & \text{for type-1 arrivals,} \\ \prod_{k=1}^{K} \prod_{i=0}^{N_k(\mathbf{n})-1} \gamma_k(i) & \text{for type-2 arrivals,} \end{cases} \tag{26}$$

where $N(\mathbf{n})$ is the total number of customers and $N_k(\mathbf{n})$ is the total number of chain-k customers in the network for network state $\mathbf{n}$. Note that if all chains are closed, $a(\mathbf{n}) = 1$ by definition. If at least one chain is open, then the product-form solution given by eqs. (25) and (26) is applicable if for each closed chain, say chain j, $\gamma_j(i)$ is set equal to zero in $\gamma(i)$ for networks with type-1 arrivals or $\gamma_j(i)$ is set equal to 1 for all i in eq. (26) for type-2 arrivals.

One way to view a closed network is that it is an open network, but the routing chain population sizes are kept fixed by two mechanisms:

(1) a loss mechanism whereby a new external arrival is discarded and lost forever;
(2) a trigger mechanism whereby a departure from the network triggers the instantaneous injection of a customer into the same chain as the departed customer (from an infinite supply of customers).

A closed network is thus equivalent to a network of open chains with the above two mechanisms in place all the time.

The above mechanisms can be invoked or revoked as a function of the population vector $\mathbf{N}$ corresponding to the current state of the network. This strategy gives rise to networks with arbitrary sets of feasible population vectors (see Figure 5). Such networks are said to have population size constraints, and it was shown by this author [8] that if V is an irreducible set of feasible population vectors, then a *sufficient condition* for the product-form solution in eqs. (25) and (26) to remain valid is: For any k, and population vectors $\mathbf{N}$ and $\mathbf{N} + \mathbf{1}_k$ in V, the loss mechanism is invoked for a chain-k external arrival in any network state with population vector $\mathbf{N}$ if and only if the trigger mechanism is invoked for a chain-k external departure in any network state with population vector $\mathbf{N} + \mathbf{1}_k$. (In other words, feasible transitions between adjacent feasible population vectors in Figure 5 are paired.)

The class of networks with population size constraints provides a general model that includes networks with closed chains, networks with open chains, and networks with mixed open and closed chains as special cases. The normalization constant G is given by the sum of the unnormalized product-form solution in eq. (25) over all feasible $\mathbf{n}$ states for each feasible population vector in the set V.

Let S denote a detailed network state that is Markovian (see [1]),

$$S = (S_1, S_2, \ldots, S_M),$$

where S_m is the state description of service center m.

Let $\mathcal{S}$ be the set of all feasible Markovian network states and $\mathcal{S}(\mathbf{N})$ be the set of feasible Markovian network states with population vector $\mathbf{N}$. Since V is the set of

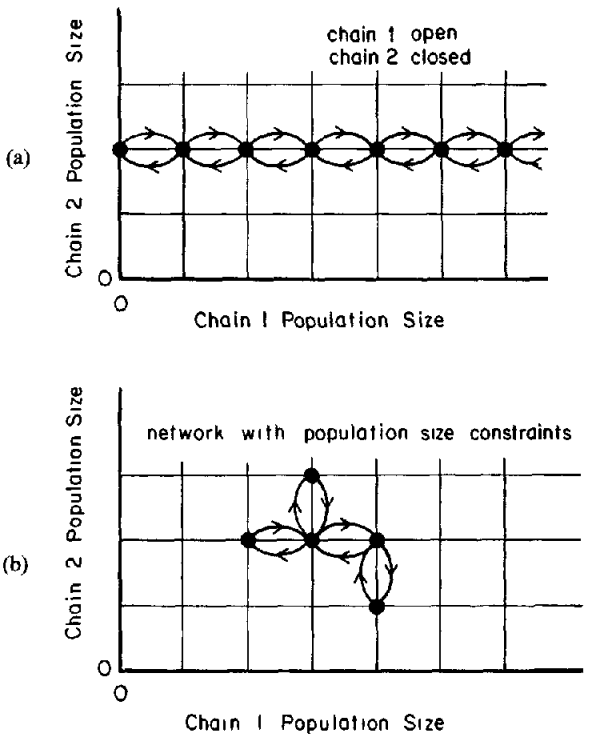

FIG 5. Examples of two-chain networks with external arrivals and departures

feasible population vectors, we have

$$\mathscr{S} = \bigcup_{\mathbf{N} \text{ in } V} \mathscr{S}(\mathbf{N}).$$

Note that $\mathscr{S}(\mathbf{N})$ is also the set of feasible states of a closed network with population vector N. We explore below the relationship between the normalization constant $G(\mathbf{N})$ of a closed network and the equilibrium probability of the aggregate state $\mathscr{S}(\mathbf{N})$ in a general network. We have found that they are also related to equilibrium state probabilities of a birth–death process traversing over the feasible population vectors in V.

It is shown in [1] that the equilibrium probability of a Markovian network state has the product form,

$$P(S) \triangleq \frac{\Pi^*(S)}{G} \triangleq \frac{a(\mathbf{N})\Pi(S)}{G}$$
$$= \frac{a(\mathbf{N})\Pi_1(S_1)\Pi_2(S_2) \cdots \Pi_M(S_M)}{G}, \tag{27}$$

where $\mathbf{N}$ is the population vector of Markovian network state S, $\Pi_m(S_m)$ is defined in [1], and

$$a(\mathbf{N}) = \begin{cases} \prod_{i=0}^{N-1} \gamma(i) & \text{for type-1 arrivals,} \\ \prod_{k=1}^{K} \prod_{i=0}^{N_k-1} \gamma_k(i) & \text{for type-2 arrivals.} \end{cases} \tag{28}$$

LOCAL BALANCE AND THE $M \Rightarrow M$ PROPERTY. Chandy [3] first observed that the product-form solution $P(S)$ of many queuing networks has a local balance property, that is, it satisfies certain local balance equations in addition to the global balance equations [1, 8]. This observation has proved to be very useful in the discovery and characterization of the class of BCMP networks [1]. (Another treatment of local balance can be found in the work of Chandy et al. [4].)

Muntz [11] found that individual service centers in BCMP networks have the $M \Rightarrow M$ property, which can be explained and related to the local balance property as follows. Consider class c in service center m (viewed in isolation). Center m has the $M \Rightarrow M$ property if given that the arrival process of customers to class c is a Poisson process, the departure process of customers from class c is also a Poisson process. Let class c be in chain k. Consider network state S in $\mathcal{S}(\mathbf{N})$. Let $\mathcal{S}^{+c}$ be the set of network states in $\mathcal{S}(\mathbf{N} + \mathbf{1}_k)$ that are the same as network state S but with an extra class-c customer in service center m. S_m is the mth component of S describing service center m. S_m^{+c} is the mth component of network state S^{+c} in $\mathcal{S}^{+c}$; it describes service center m with the extra class-c customer. $\Pi_m(S_m)$ in the product-form solution was found to satisfy the following sufficient condition for the $M \Rightarrow M$ property [11]:

$$\sum_{S_m^{+c} \text{ in } \mathcal{S}_m^{+c}} \frac{\Pi_m(S_m^{+c}) R_m(S_m^{+c} \to S_m)}{\Pi_m(S_m)} = \nu_c, \tag{29}$$

where ν_c was defined in eq. (23), $\mathcal{S}_m^{+c}$ is the set of center-m components of network states in $\mathcal{S}^{+c}$, and $R_m(S_m^{+c} \to S_m)$ is the transition rate from S_m^{+c} to S_m corresponding to the departure of the extra class-c customer from center m. Equation (29) can be rewritten as

$$\Pi_m(S_m)\nu_c = \sum_{S_m^{+c} \text{ in } \mathcal{S}_m^{+c}} \Pi_m(S_m^{+c}) R_m(S_m^{+c} \to S_m), \tag{30}$$

where we can interpret

(a) the left-hand side of eq. (30) to be the "flow" out of state S_m due to class-c arrivals, and
(b) the right-hand side of eq. (30) to be the flow into state S_m due to class-c departures.

Equation (30) is an example of a local balance equation. Since it is with respect to the arrivals and departures of a specific class, it will be referred to as a *class local balance* equation [1, 3].

Since $\Pi(S)$ has a product form, the previous equation can be rewritten as

$$\Pi(S)\nu_c = \sum_{S^{+c} \text{ in } \mathcal{S}^{+c}} \Pi(S^{+c}) R_m(S_m^{+c} \to S_m). \tag{31}$$

We next employ eq. (31) to demonstrate a local balance property of $\Pi^*(S)$ with respect to external arrivals and departures of a routing chain; this will be referred to as *chain local balance*. Consider chain k. Suppose the population vectors $\mathbf{N}$ and $\mathbf{N} + \mathbf{1}_k$ are in V with transitions between them allowed.

The following identity,

$$1 = \sum_{c \text{ in } \mathrm{RC}(k)} \nu_c \left[1 - \sum_{d \text{ in } \mathrm{RC}(k)} p_{cd} \right], \tag{32}$$

can be easily demonstrated using eq. (23) and $\sum_{c \text{ in } \mathrm{RC}(k)} q_c = 1$. Now replace ν_c in the

right-hand side of the above using eq. (31) and get

$$1 = \sum_{c \text{ in } \mathrm{RC}(k)} \frac{\sum_{S^{+c} \text{ in } \mathscr{S}^{+c}} \Pi(S^{+c}) R_m(S_m^{+c} \to S_m)}{\Pi(S)} \left[1 - \sum_{d \text{ in } \mathrm{RC}(k)} p_{cd} \right].$$

Let $\mathbf{N}$ be the population vector of network state S, and define

$$\gamma_k(\mathbf{N}) = \begin{cases} \gamma_k(N) & \text{for type-1 arrivals,} \\ \gamma_k(N_k) & \text{for type-2 arrivals.} \end{cases} \tag{33}$$

Multiplying both sides of the previous equation by $\gamma_k(\mathbf{N})$ and rewriting $\Pi(S^{+c})/\Pi(S)$ as $\Pi^*(S^{+c})/[\gamma_k(\mathbf{N})\Pi^*(S)]$, we get

$$\Pi^*(S)\gamma_k(\mathbf{N}) = \sum_{c \text{ in } \mathrm{RC}(k)} \sum_{S^{+c} \text{ in } \mathscr{S}^{+c}} \Pi^*(S^{+c}) R_m(S_m^{+c} \to S_m) \left[1 - \sum_{d \text{ in } \mathrm{RC}(k)} p_{cd} \right], \tag{34}$$

which then is a local balance equation satisfied by $\Pi^*(S)$ with respect to chain k; note that $[1 - \sum_{d \text{ in } \mathrm{RC}(k)} p_{cd}]$ is the probability that the extra class-c customer departing from center m leaves the network instead of joining another service center. We can interpret

(a) the left-hand side of eq. (34) to be the flow out of state S due to chain-k arrivals, and

(b) the right-hand side of eq. (34) to be the flow into state S due to chain-k departures.

Note that eq. (34) is applicable only if transitions between $\mathbf{N}$ and $\mathbf{N} + \mathbf{1}_k$ are feasible. We have thus shown the following lemma.

LEMMA 3. *The class local balance property of* $\Pi_m(S_m)$ *in the product-form solution* $P(S)$ *implies that* $P(S)$ *satisfies the chain local balance equation* (34).

The chain local balance property of the product-form solution is the key for demonstrating its applicability to the extended class of BCMP networks with population size constraints in [8]. It also has the following immediate consequence.

LEMMA 4 ($M \Rightarrow M$ PROPERTY FOR A ROUTING CHAIN). *If external arrivals to chain k form a Poisson process with a constant rate* γ_k, *then chain-k customers departing from the network form a Poisson process at the same rate.*

The above lemma is easily proved using eq. (34) and Muntz's arguments [11]. Note that any subnetwork of $M \Rightarrow M$ service centers will have the $M \Rightarrow M$ property with respect to each chain's external arrivals to the subnetwork and departures from the subnetwork. Hence the subnetwork behaves like a single (composite) $M \Rightarrow M$ service center to the rest of the network. (This observation, however, does not apply to networks with the third form of state-dependent service rates if the subnetwork and the set I of service centers overlap partially.)

AGGREGATE STATES AND THEIR OCCUPANCY STATISTICS. Let $P(\mathbf{N})$ denote the equilibrium probability of the aggregate state $\mathscr{S}(\mathbf{N})$, defined to be

$$P(\mathbf{N}) = \sum_{S \text{ in } \mathscr{S}(\mathbf{N})} P(S).$$

Let $S(t)$ denote the network state at time t. Consider the case of $t \to \infty$. We define the *conditional throughput rate* of chain k as

$$T_k^*(\mathbf{N} + \mathbf{1}_k) = \lim_{\Delta \to 0} \frac{1}{\Delta} P[S(t) \text{ in } \mathscr{S}(\mathbf{N}) \mid S(t - \Delta) \text{ in } \mathscr{S}(\mathbf{N} + \mathbf{1}_k)] \tag{35}$$

The next theorem characterizes the occupancy statistics of the aggregate states of a network with population size constraints and relates them to normalization constants of networks which are identical to the given network except that their chains are closed (to be referred to as equivalent closed networks).

THEOREM

(i) The equilibrium aggregate state probabilities are given by

$$P(\mathbf{N}) = \frac{a(\mathbf{N})}{G} G(\boldsymbol{\alpha}, \mathbf{N}) \qquad \text{for} \quad \mathbf{N} \quad \text{in} \quad V, \tag{36}$$

where $a(\mathbf{N})$ was defined in eq. (28), $G(\boldsymbol{\alpha}, \mathbf{N})$ is the normalization constant of an equivalent closed network with population vector $\mathbf{N}$ and scaling factors $\alpha_k = \lambda_{1k}$, $k = 1, 2, \ldots, K$, given by eqs. (23) and (24), and

$$G = \sum_{\mathbf{N} \text{ in } V} a(\mathbf{N}) G(\boldsymbol{\alpha}, \mathbf{N}). \tag{37}$$

(ii) If $\mathbf{N}$ and $\mathbf{N} + \mathbf{1}_k$ are in V with transitions permitted between them, then the conditional throughput rate is given by

$$T_k^*(\mathbf{N} + \mathbf{1}_k) = \frac{G(\boldsymbol{\alpha}, \mathbf{N})}{G(\boldsymbol{\alpha}, \mathbf{N} + \mathbf{1}_k)}, \tag{38}$$

and $P(\mathbf{N})$ satisfies the chain local balance equation,

$$P(\mathbf{N})\gamma_k(\mathbf{N}) = P(\mathbf{N} + \mathbf{1}_k) T_k^*(\mathbf{N} + \mathbf{1}_k). \tag{39}$$

PROOF. To show part (i) of the theorem, consider the improper aggregate state probability,

$$\pi(\mathbf{N}) = \sum_{S \text{ in } \mathcal{S}(\mathbf{N})} \Pi^*(S) = a(\mathbf{N}) \sum_{S \text{ in } \mathcal{S}(\mathbf{N})} \Pi(S).$$

Since $\Pi(S)$ is the (improper) product-form solution of a closed network [1] with scaling factors $\alpha_k = \lambda_{1k}$, $k = 1, 2, \ldots, K$, and $\mathcal{S}(\mathbf{N})$ is the set of feasible network states with population vector $\mathbf{N}$, we have

$$\pi(\mathbf{N}) = a(\mathbf{N}) G(\boldsymbol{\alpha}, \mathbf{N}) \qquad \text{for} \quad \mathbf{N} \quad \text{in} \quad V.$$

Normalizing these improper probabilities to sum to one, eqs. (36) and (37) immediately follow.

To show eq. (38) in part (ii) of the theorem, we rewrite eq. (35) as

$$T_k^*(\mathbf{N} + \mathbf{1}_k) = \lim_{\Delta \to 0} \frac{1}{\Delta} \frac{P[S(t - \Delta) \text{ in } \mathcal{S}(\mathbf{N} + \mathbf{1}_k) \text{ and } S(t) \text{ in } \mathcal{S}(\mathbf{N})]}{P[S(t - \Delta) \text{ in } \mathcal{S}(\mathbf{N} + \mathbf{1}_k)]}.$$

Taking the limit $\Delta \to 0$ and multiplying both numerator and demoninator by G, we have

$$T_k^*(\mathbf{N} + \mathbf{1}_k) = \frac{a(\mathbf{N} + \mathbf{1}_k) \sum_{c \text{ in } \mathrm{RC}(k)} \sum_{S \text{ in } \mathcal{S}(\mathbf{N})} \sum_{S^{+c} \text{ in } \mathcal{S}^{+c}} \Pi(S^{+c}) R_m(S_m^{+c} \to S_m)[1 - \sum_{d \text{ in } \mathrm{RC}(k)} p_{cd}]}{\pi(\mathbf{N} + \mathbf{1}_k)},$$

where

$$\pi(\mathbf{N} + \mathbf{1}_k) = a(\mathbf{N} + \mathbf{1}_k) G(\boldsymbol{\alpha}, \mathbf{N}).$$

Cancel the term $a(\mathbf{N} + \mathbf{1}_k)$ in both the numerator and denominator and note that the expression in the numerator,

$$\sum_{S \text{ in } \mathscr{S}(\mathbf{N})} \sum_{S^{+c} \text{ in } \mathscr{S}^{+c}} \Pi(S^{+c}) R_m(S_m^{+c} \to S_m),$$

divided by $G(\alpha, \mathbf{N} + \mathbf{1}_k)$, is by definition equal to the throughput rate of class-c customers in an equivalent closed network with population vector $\mathbf{N} + \mathbf{1}_k$ and scaling factors α, which is

$$T_c(\mathbf{N} + \mathbf{1}_k) = \frac{v_c G(\alpha, \mathbf{N})}{G(\alpha, \mathbf{N} + \mathbf{1}_k)}.$$

We then have

$$\begin{aligned} T_k^*(\mathbf{N} + \mathbf{1}_k) &= \sum_{c \text{ in } \mathrm{RC}(k)} T_c(\mathbf{N} + \mathbf{1}_k) \left[1 - \sum_{d \text{ in } \mathrm{RC}(k)} p_{cd} \right] \\ &= \sum_{c \text{ in } \mathrm{RC}(k)} \frac{G(\alpha, \mathbf{N})}{G(\alpha, \mathbf{N} + \mathbf{1}_k)} v_c \left[1 - \sum_{d \text{ in } \mathrm{RC}(k)} p_{cd} \right] \\ &= \frac{G(\alpha, \mathbf{N})}{G(\alpha, \mathbf{N} + \mathbf{1}_k)}, \end{aligned}$$

which is eq. (38) in which we have made use of the identity in (32).

Equation (39) is a consequence of eqs. (28), (36), and (38). It can also be shown by summing the chain local balance equation (34) over S in $\mathscr{S}(\mathbf{N})$ and recognizing that the resulting equation is

$$GP(\mathbf{N})\gamma_k(\mathbf{N}) = GT_k^*(\mathbf{N} + \mathbf{1}_k)P(\mathbf{N} + \mathbf{1}_k). \qquad \square$$

We can interpret eq. (39) as a *chain local balance* equation satisfied by $P(\mathbf{N})$, since it equates the flow out of the aggregate state $\mathscr{S}(\mathbf{N})$ due to chain-k arrivals to the flows into $\mathscr{S}(\mathbf{N})$ due to chain-k departures.

Let us relate the conditional throughput rate in eq. (38) to previous results. Recall that the throughput rate of a *closed* network with population vector $\mathbf{N} + \mathbf{1}_k$ is defined to be the throughput rate of service center 1, which is arbitratily chosen. It is given by

$$T_k(\mathbf{N} + \mathbf{1}_k) = \alpha_k \frac{G(\alpha, \mathbf{N})}{G(\alpha, \mathbf{N} + \mathbf{1}_k)},$$

where α_k is equal to the *relative* arrival rate λ_{1k} of chain-k customers to center 1. The throughput rate of chain-k customers at service center m is given by $(\lambda_{mk}/\lambda_{1k})T_k(\mathbf{N} + \mathbf{1}_k)$.

Consider chain k which permits external arrivals and departures. Note that λ_{mk} given by eqs. (23) and (24) can be interpreted as the mean number of visits by a chain-k customer to service center m between successive visits to a service center outside the network introduced to act as the source and sink of chain-k customers. For an "open" chain it is physically meaningful to define its throughput rate to be that of its source/sink center. With the set of relative arrival rates defined in eqs. (23) and (24), the relative arrival rate to the source/sink center is unity. Hence

$$T_k^*(\mathbf{N} + \mathbf{1}_k) = \frac{1}{\lambda_{1k}} T_k(\mathbf{N} + \mathbf{1}_k),$$

which is eq. (38).

We make the following additional observations.

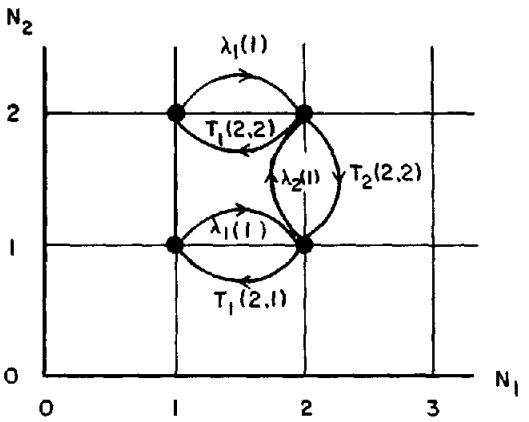

FIG. 6. An example of a two-chain network with population size constraints.

COROLLARY

(i) The equilibrium aggregate state probabilities are the same as the equilibrium state probabilities of a birth–death process with state space V, birth rates $\gamma(\mathbf{N})$, *and death rates* $T_k^*(\mathbf{N} + \mathbf{1}_k)$ *for* $\mathbf{N}$ *and* $\mathbf{N} + \mathbf{1}_k$ *in V.*

(ii) The equilibrium aggregate state probabilities are independent of feasible transitions in V imposed by the loss and trigger mechanisms.

Part (ii) of the corollary is obvious from Lemma 2. It also implies that $P(S)$ is independent of feasible transitions in V. It does, however, depend upon the set V through the normalization constant G.

AN EXAMPLE. Consider a network with two chains. The set V of feasible population vectors consists of (1, 1), (2, 1), (1, 2), and (2, 2). Type-2 arrival processes are assumed. The feasible transitions in V are shown in Figure 6.

Instead of applying eq. (36), we shall solve for $P(N_1, N_2)$ directly using the local balance eq. (39), from which we get the relationships[1],

$$P(2, 1) = \frac{\lambda_1(1)}{T_1(2, 1)} P(1, 1),$$

$$P(2, 2) = \frac{\lambda_2(1)}{T_2(2, 2)} P(2, 1),$$

$$P(2, 2) = \frac{\lambda_1(1)}{T_1(2, 2)} P(1, 2).$$

Letting $P(1, 1) = C$ and solving for the others in terms of C, we get

$$P(1, 1) = C,$$

$$P(2, 1) = \frac{\lambda_1(1)}{T_1(2, 1)} C,$$

$$P(2, 2) = \frac{\lambda_2(1)\lambda_1(1)}{T_2(2, 2)T_1(2, 1)} C,$$

$$P(1, 2) = \frac{T_1(2, 2)}{\lambda_1(1)} \frac{\lambda_2(1)\lambda_1(1)}{T_2(2, 2)T_1(2, 1)} C.$$

[1] For simplicity we have omitted the * notation from T_1 and T_2.

Applying Lemma 2 to the two paths of increasing sequences of population vectors from (1, 1) to (2, 2), we have

$$T_2(1, 2)T_1(2, 2) = T_2(2, 2)T_1(2, 1).$$

We can then rewrite the solution for $P(1, 2)$ as

$$P(1, 2) = \frac{\lambda_2(1)}{T_2(1, 2)}C.$$

The constant C can then be determined from

$$P(1, 1) + P(2, 1) + P(1, 2) + P(2, 2) = 1.$$

EVALUATION OF THE NORMALIZATION CONSTANT G. The normalization constant G in eq. (37) is evaluated as a summation over the set V of feasible population vectors. For open chains without population size constraints the set V is infinite. If the external arrival rates to the open chains are constants, that is,

$$\gamma_k(\mathbf{N}) = \gamma_k,$$

then G can be found easily. First, if all chains in the network are open, then it is well known [1] that

$$G = \prod_{m=1}^{M} \frac{1}{1 - \rho_m}, \qquad \text{where} \quad \rho_m = \sum_k \rho_{mk}.$$

Second, if some of the chains in the network are open while the rest are closed, then it has been shown [14] that

$$G = G_{\text{open}} \cdot G(\mathbf{N}),$$

where

$$G_{\text{open}} = \prod_{m=1}^{M} \frac{1}{1 - \rho_m^{\text{o}}}, \qquad \rho_m^{\text{o}} = \sum_{k \text{ open}} \rho_{mk}.$$

The normalization constant for the closed chains with population vector $\mathbf{N}$ can then be evaluated separately with some modifications to account for interactions (if any) between open and closed chains at individual service centers. Let

$$\rho_m^{\text{c}} = \sum_{k \text{ closed}} \rho_{mk}.$$

(1) At an IS center, open and closed chains do not interact. No modification is necessary in the computation of $G(\mathbf{N})$ with respect to the IS center.

(2) At a fixed-rate center the closed-chain traffic intensity should be modified as follows in the computation of $G(\mathbf{N})$:

$$\rho_m^{\text{c}} \leftarrow \frac{\rho_m^{\text{c}}}{1 - \rho_m^{\text{o}}},$$

to account for the effect of the open chains on the closed chains at this center.

(3) At a queue-dependent-service-rate center, the interactions are more complex than the above, and the effect of the open-chain traffic intensity ρ_m^{o} needs to be accounted for by a convolution operation (see [14]).

If the chain arrival rates $\gamma_k(\mathbf{N})$ depend upon the population vector and/or the network has population size constraints, then G must be evaluated from eq. (37), repeated here:

$$G = \sum_{\mathbf{N} \text{ in } V} a(\mathbf{N})G(\alpha, M, \mathbf{N}).$$

Note that all normalization constants $G(\alpha, M, \mathrm{N})$ of the equivalent closed networks must use the same set of scaling factors. Hence it is likely that no single set of scaling factors can be found so that $G(\alpha, M, \mathrm{N})$, N in V, will fit into a given range of floating-point numbers. Since we are dealing with a summation of terms, if some terms in the sum are too small relative to the others (i.e., underflow occurs), they can be discarded. The error introduced in G is negligible if $|V|$ SMALLEST $\ll$ LARGEST, where $|V|$ denotes the cardinality of V.

5. *Conclusions*

We have found that previous difficulties with evaluating the normalization constants of closed BCMP queuing networks are due to the use of a fixed set of scaling factors. Normalization constants $G(\alpha, M, \mathrm{N})$ and $G(\beta, M, \mathrm{N})$ based upon different scaling factors were found to be related very simply by

$$G(\alpha, M, \mathrm{N}) = \prod_{k=1}^{K} \left(\frac{\alpha_k}{\beta_k}\right)^{N_k} G(\beta, M, \mathrm{N}).$$

As a result, in the course of evaluating a set of normalization constants (using any computational algorithm) one can repeatedly change the set of scaling factors to avoid overflow or underflow problems that might be encountered. Hence normalization constants for very large population sizes can be obtained with computers having just a modest range of floating-point numbers.

The MVA algorithm of Reiser and Lavenberg [15] bypasses normalization constants and computes various network performance measures directly. It is sometimes desirable to solve a queuing network problem using a hybrid solution method that employs more than one computational algorithm, for example, MVA for fixed-rate servers and convolution for queue-dependent servers. In this case normalization constants are needed and can be computed from the outputs of the MVA algorithm, using eq. (13), for example. Dynamic scaling will then be necessary.

We have also considered BCMP networks with external arrivals, departures, and population size constraints. We have shown that the class local balance property possessed by the product-form solution implies several interesting properties for chains. In particular, external arrivals to a chain and departures from the chain are characterized by the $M \Rightarrow M$ property. The relationships between normalization constants of closed networks and equilibrium aggregate state probabilities of networks that permit external arrivals and departures have been examined. We have found that the growth behavior of normalization constants can be modeled by a birth–death process traversing over the set of population vectors.

To show that the results presented in this paper (Lemmas 1–4 and the theorem) are applicable to networks with any or all three forms of state-dependent service rates allowed in BCMP networks, we note that the class local balance equation in eq. (31) is valid for all three forms of state-dependent service rates. Note also that Lemmas 1 and 2 are based on eq. (12), which may be derived from eq. (31). Lemmas 3 and 4 and the theorem are all based on eq. (31).

ACKNOWLEDGMENTS. The author thanks Herbert Schwetman of Purdue University who, as an editor of *Communications of the ACM*, handled the initial reviewing process of this paper. This paper has benefited from comments on the original manuscript [9] by Peter Denning of Purdue University, and Steve Lavenberg and Charles Sauer of the IBM Thomas J. Watson Research Center. The constructive criticisms of the anonymous referees are also greatly appreciated.

REFERENCES

1. BASKETT, F., CHANDY, K.M., MUNTZ, R.R., AND PALACIOS, F. Open, closed, and mixed networks of queues with different classes of customers. *J. ACM 22*, 2 (April 1975), 248–260.
2. BUZEN, J.P. Computational algorithms for closed queuing networks with exponential servers. *Commun. ACM 16*, 6 (Sept. 1973), 527–531.
3. CHANDY, K.M. The analysis and solutions for general queuing networks. Proc. 6th Ann. Princeton Conf. on Information Sciences and Systems, Princeton, N.J., 1972, pp. 224–228.
4. CHANDY, K.M., HOWARD, J.H., AND TOWSLEY, D.F. Product form and local balance in queuing networks. *J. ACM 24*, 2 (April 1977), 250–263.
5. CHANDY, K.M., AND SAUER, C.H. Computational algorithms for product-form queuing networks. *Commun. ACM 23*, 10 (Oct. 1980), 573–583.
6. CHANG, A., AND LAVENBERG, S.S. Work rates in closed queuing networks with general independent servers. *Oper. Res. 22*, 4 (1974), 838–847.
7. JACKSON, J.R. Jobshop-like queuing systems. *Manage. Sci. 10* (Oct. 1963), 131–142.
8. LAM, S.S. Queuing networks with population size constraints. *IBM J. Res. Devel. 21*, 4 (July 1977), 370–378.
9. LAM, S.S. Behavior of the normalization constant and a scaling technique for product-form queueing networks. Tech. Rep. TR-148, Dep. of Computer Sciences, Univ. of Texas, Austin, Texas, June 1980.
10. LAM, S.S., AND LIEN, Y.L. A tree convolution algorithm for the solution of queueing networks. Tech. Rep. TR-165, Dep. of Computer Sciences, University of Texas, Austin, Texas, Jan. 1981.
11. MUNTZ, R.R. Poisson departure process and queuing networks. Proc. 7th Ann. Princeton Conf. on Information Sciences and Systems, Princeton, N.J., March 1973, 435–440.
12. MUNTZ, R.R., AND WONG, J.W. Asymptotic properties of closed queuing network models. Proc. 8th Ann. Princeton Conf. on Information Sciences and Systems, Princeton, N.J., March 1974, 348–352.
13. REISER, M. Numerical methods in separable queuing networks. *Studies Manage. Sci. 7* (1977), 113–142.
14. REISER, M., AND KOBAYASHI, H. Queuing networks with multiple closed chains: Theory and computational algorithms. *IBM J. Res. Devel. 19*, 3 (May 1975), 283–294.
15. REISER, M., AND LAVENBERG, S.S. Mean-value analysis of closed multichain queuing networks. *J. ACM 27*, 2 (April 1980), 313–322.

RECEIVED JULY 1980, REVISED DECEMBER 1980; ACCEPTED DECEMBER 1980