BEHAVIOR OF THE NORMALIZATION CONSTANT AND A SCALING TECHNIQUE FOR PRODUCT-FORM QUEUEING NETWORKS*

Simon S. Lam

TR-148 June 1980

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

*This work was supported by National Science Foundation Grant No. ENG78-01803.

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 queueing networks. Additional time and space overheads needed by the technique are found to be minimal. 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. Product-form queueing networks with external arrivals, departures and population size constraints are also considered. It is shown that the network population vector is characterized by a continuous-time Markov chain. The equilibrium probabilities of feasible population vectors are related to normalization constants of equivalent closed networks.

1. INTRODUCTION

Queueing networks have been used extensively and successfully in the modeling of computer systems and communication networks. Jackson [1] 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 [2,3,4]. 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-from solution was later shown to be also applicable to an extended class of BCMP networks with arbitrary constraints on chain population sizes [5].

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.

Several computational algorithms are available for the class of BCMP networks [6,7,8,9]. The convolution algorithm was first discovered by Buzen [6] for single-chain networks and extended by Reiser and Kobayashi [7] to multi-chain networks. The LBANC and CCNC algorithms were recently proposed by Chandy and Sauer [9]. These algorithms all attempt to first evaluate the normalization constnats 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 $G(\underline{N})$ of a network with population vector $\underline{N}$ using any of these algorithms is that as the chain population sizes in $\underline{N}$ become large, $G(\underline{N})$ may become too large (causing a floating point overflow) or too small (causing a floating point underflow) [9,10]. A scaling technique was described by Reiser [10] 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 Lavenburg [8] bypasses the evaluation of $G(\underline{N})$ and computes various network performance measures directly. However, Chandy and Sauer [9] found that the MVA algorithm may encounter some other difficulties and may have unstable numerical characteristics under certain conditions.

Summary of our results

The overflow and underflow problems encountered in the evaluation of $G(\underline{N})$ using current algorithm implementations are due to the use of a fixed set of "scaling factors" for the entire range of values of $\underline{N}$ of interest. We found that the scaling factors can be factored out of the expression for $G(\underline{N})$ so that one can just as easily use different sets of scaling factors for different values of $\underline{N}$ with just small amounts of space and computation overheads. As a result, the scaling factors can be changed to smaller values when $G(\underline{N})$ is about to encounter an overflow, and changed to larger values when $G(\underline{N})$ 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 $G(\underline{N})$ for a wide range of values of $\underline{N}$ using computers with a small floating point range or even computers without floating point numbers! The scaling technique and related results are covered in Section III below.

External Poisson arrivals at rates that may depend upon routing chain population sizes are allowed in BCMP networks [3] and the extended class of BCMP networks with population size constraints [5]. In these networks, the population vector $\underline{N}$ can have more than one set of values.
We found that the population vector $\underline{N}$ can be characterized as a continuous-time Markov chain. The normalization constant of such a network is obtainable from the normalization constants of equivalent closed networks over the space of feasible population vectors. These results are covered in Section IV below.

II DEFINITIONS AND NOTATIONS

Service centers in a network 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 $SC(m)$ 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 and only if c and d are in different chains. Moreover, each chain is irreducible i.e., the transition probabilities $\{p_{cd}$; c,d in 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,...,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

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

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

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

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

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

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

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

We define the nominal traffic intensity to be

$$w_{mk} = \lambda_{mk} \tau_{mk} \qquad \text{for } \alpha_k = 1 \qquad (5)$$

Thus, we have

$$\rho_{mk} = \alpha_k w_{mk} \qquad (6)$$

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 later in Section IV.) We let N_k be the number of customers in chain k. The network population vector is

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

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

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

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

where

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

(We note that $\underline{n}$ 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 $\underline{N}$ is [3]

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

where

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

where

$$n_m = n_{m1} + n_{m2} + \ldots + n_{mK}$$

The form of (8) is the same for all 4 types of service centers considered in [3]; 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 i.e., $\tau_c = \tau_m$ 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

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

III. CLOSED NETWORKS

Examining Equations (8) and (9), we note that $G(\underline{N})$ is a function of $\underline{N}$, 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

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

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

Lemma 1

$$G(\underline{\alpha},M,\underline{N}) = \alpha_1^{N_1} \alpha_2^{N_2} \ldots \alpha_K^{N_K} \quad G(\underline{1},M,\underline{N}) \qquad (10)$$

where $\underline{1}$ 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

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

where

$$r(\underline{\beta},\underline{\alpha},\underline{N}) = \prod_{k=1}^{K} (\beta_k/\alpha_k)^{N_k}$$

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

It is instructional, 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 $\underline{N}$ is given by [6,7,9]

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

where $\underline{1}_k$ is a K-vector with the k^{th} 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. (12) can be rewritten as

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

A consequence of (12) is that the ratio $\lambda_{mk}/T_{mk}(\underline{N})$ 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 $T_k(\underline{N})$ for

$T_{1k}(\underline{N})$. The above equation can now be rewritten as

$$G(\underline{N}) = \frac{\alpha_k}{T_k(\underline{N})} \quad G(\underline{N} - \underline{1}_k) \qquad \underline{N} \geq \underline{1}_k \tag{13}$$

Traditionally, we first compute $G(\underline{N})$ and then derive $T_k(\underline{N})$ from $G(\underline{N})$ and $G(\underline{N} - \underline{1}_k)$. Now since we are interested in the behavior of $G(\underline{N})$, we consider the reverse process. Note that $T_k(\underline{N})$ can be obtained from the MVA algorithm directly and is independent of the scaling factor α_k [8].

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

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

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

$$\begin{aligned}
\underline{N}^{(0)} &= \underline{0} \\
\underline{N}^{(1)} &= \underline{N}^{(0)} + \underline{1}_{k_1} \\
\underline{N}^{(2)} &= \underline{N}^{(1)} + \underline{1}_{k_2} \\
&\cdot \\
&\cdot \\
&\cdot \\
\underline{N}^{(N)} &= \underline{N}^{(N-1)} + \underline{1}_{k_N} = \underline{N}
\end{aligned}$$

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

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

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

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

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

$$\prod_{i=1}^{N} T_{k_i}(\underline{N}^{(i)}) = \text{constant} \tag{15}$$

Let us set aside the above result until Section IV. We shall now consider the special case of $K = 1$ i.e., 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 afterwards.

For a network with a single closed chain our previous notation will be 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

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

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 shall assume for the moment that service rate functions are limited to the following

$$\mu_m(i) = \begin{cases} i & 1 \leq i \leq j_m \\ j_m & i \geq j_m \end{cases} \tag{17}$$

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 [11] 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,...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 (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, i.e.

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

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

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

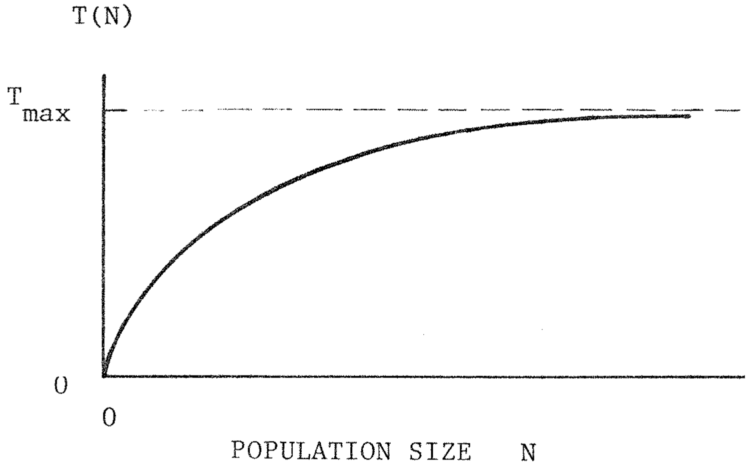

Fig. 1. Throughput rate versus population size in a single-chain network.

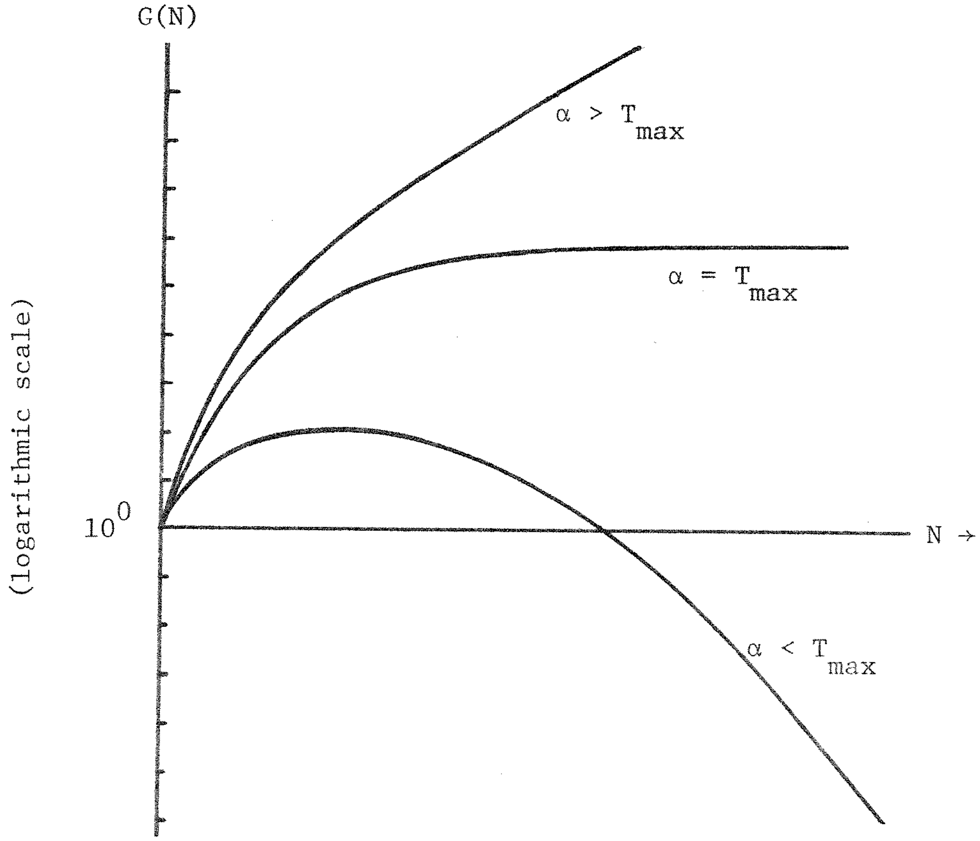

Fig. 2. Behavior of G(N) in a single-chain network.

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

$$T(N) \leq \frac{u_1}{u_{m^*}} \frac{j_1}{\tau_1} \overset{\text{def.}}{=} T_{max} \tag{18}$$

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

Referring back to (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 3 general cases of behavior are illustrated in Figure 2. We see that if $\alpha \geq T_{max}$ 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

In current computational algorithm implementations the same scaling factor α is used to compute G(N) for the full range of N values of interest. Lemma 1 and (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 scaling technique, we use an example considered by Chandy and Sauer in [9] and is illustrated in Figure 3. Center 4 is an IS center that models a population of terminals. Centers 1, 2 and 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	0.4	0.008
4	0.2	15

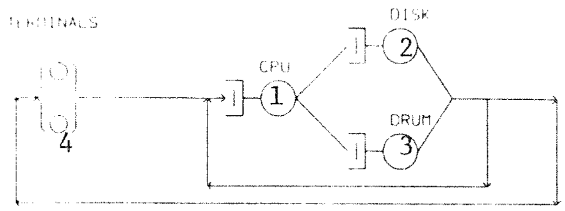

Fig. 3. Single-chain network example.

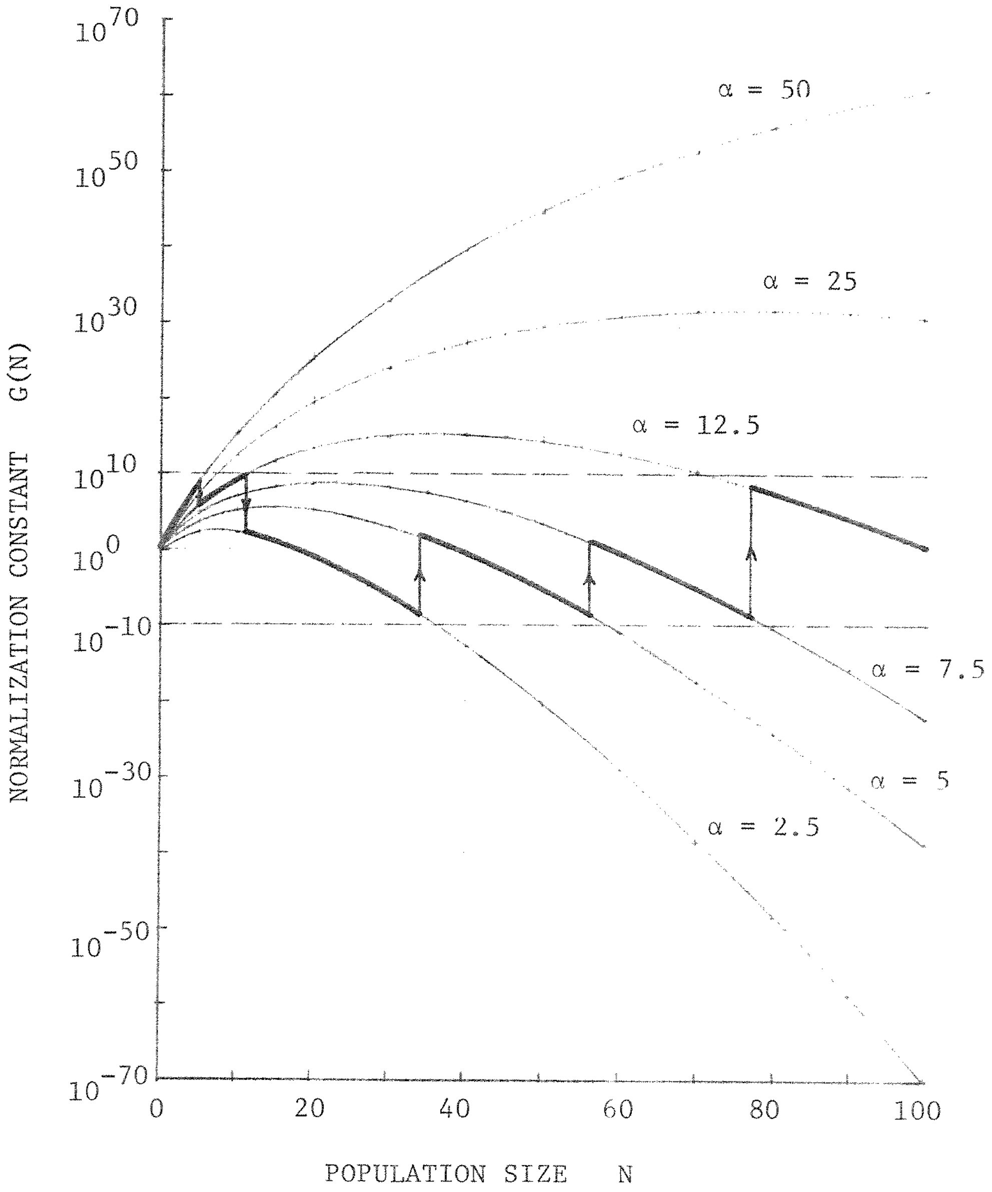

Fig. 4. Dynamic scaling for single-chain network example.

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 (11). When a floating 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} to 10^{10} floating point range. It is not unlikely that we ended up with a scaling factor that we used earlier. But the scaling technique enabled us to 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 and in general $T(\underline{N} + \underline{1}_k)$ is not necessarily larger than $T(\underline{N})$. 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 3 fixed-rate centers with 2 routing chains. The nominal traffic intensities w_{mk} (for $\alpha_1 = \alpha_2 = 1$) are

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

Let us employ the convolution algorithm for fixed-rate servers from [7]. Let $G(\underline{\alpha},m,\underline{N})$ denote the normalization constant for the first m centers with scaling factors $\underline{\alpha}$ and population vector $\underline{N}$. We have

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

and

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

The above recursive equation can be rewritten as

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

where $r(\underline{\alpha},\underline{\beta},\underline{N})$ was defined earlier. Suppose in the 2-chain network example we want the normalization constant for $\underline{N} = (2,2)$. However, the largest value of the normalization constant that we can store is 100. By dynamically changing the scaling factors and employing (19) we arrived at the results tabulated in Table 1 below.

	(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
	$\underline{\alpha}$	(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	$30\frac{4}{9}$
	$\underline{\alpha}$	(1,1)	(1,1)	(1,1)	(1,1)	(1,1)	(1,1)	$(\frac{1}{3},\frac{1}{2})$	$(\frac{1}{3},\frac{1}{2})$	$(\frac{1}{6},\frac{1}{2})$
m=3	G	1	7	8	78	35	44	$21\frac{1}{6}$	$7\frac{7}{9}$	$41\frac{7}{18}$
	$\underline{\alpha}$	(1,1)	(1,1)	(1,1)	(1,1)	(1,1)	(1,1)	$(\frac{1}{6},\frac{1}{2})$	$(\frac{1}{6},\frac{1}{2})$	$(\frac{1}{6},\frac{1}{2})$

Table 1. Normalization constants and their scaling factors for the two-chain network example.

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.

The computation of service center throughput rates can be done using the formula

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

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

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

where G_{m+} is the output of the convolution algorithm over centers 1 - M but with center m convolved twice [7]. 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 with $\mu_m(i)$ not being a constant. In this case, the marginal queue length distribution may need to be first computed as follows

$$P_m(\underline{n}_m) = \frac{p_m(\underline{n}_m) \quad G_{m-}(\underline{\alpha},M,\underline{N} - \underline{n}_m)}{r(\underline{\alpha},\underline{\beta},\underline{N}) \; G(\underline{\beta},M,\underline{N})}$$

where $p_m(\underline{n}_m)$ was defined earlier G_{m-} is the output of the convolution algorithm over centers 1 - M but skipping over center m. Since $\underline{n}_m$ may range from $\underline{0}$ to $\underline{N}$, it will then be likely that we cannot fit the normalization constants of $\underline{N} - \underline{n}_m$ and $\underline{N}$ 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 $P_m(\underline{n}_m)$ that are too small and will cause underflows!

Time and space overheads

The additional time overhead of dynamic scaling is rather insignificant. Each time the scaling factors are changed, (11) needs to be computed. Assuming that the available floating point range is not too small and $G(\underline{N})$ does not fluctuate greatly as a function of $\underline{N}$ (due to fluctuations in $\mu_m(i)$), the frequency of encountering overflow or underflow conditions requiring a change in scaling factors, should be very low.

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 $\underline{0}$ and $\underline{N}$ is needed. A straight-forward way to provide a mapping between population vectors and their corresponding scaling factors is to provide an entire array of $\underline{\alpha}$ values. However, an inspection of the example in Table 1 suggests that since changes occur infrequently the mapping between population vectors and scaling factors can be easily accomplished with some appropriate data structures; a substantial saving in storage requirement may be achieved. With LBANC and MVA algorithms, since the recursion is done over the population vectors, additional saving is possible since an entire array, indexed from $\underline{0}$ to $\underline{N}$, of normalization constants is not needed.

The least amount of space overhead needed for dynamic scaling, with any computational algorithm, is to use a single scaling factor, say α , for all chains (at the expense of, perhaps, some flexibility). This way, only the mapping between N ($=N_1 + N_2 + \ldots + N_K$) 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.

A simple technique to do scaling is as follows. Let $G(\alpha,M,\underline{N})$ 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 $\underline{N}$ as follows

$$\alpha \leftarrow \beta\,\alpha$$

$$G \leftarrow \beta^{N}\,G$$

where $N = N_1 + N_2 + \ldots + N_K$. Suppose LARGE and SMALL denote the largest and smallest numbers that we can use. To scale G down to about unity when an overflow occurs, we can choose

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

To scale G up to about unity when an underflow occurs, we can choose

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

IV. NETWORKS WITH EXTERNAL ARRIVALS, DEPARTURES AND POPULATION SIZE CONSTRAINTS

The queueing network model described in Section II is for closed routing chains, each with a fixed number of circulating customers. The model is next 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

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

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 (1))

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

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

There can be 2 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 $\gamma(N) = \gamma_1(N) + \gamma_2(N) + \ldots + \gamma_K(N)$

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 open and closed chains, Baskett, Chandy, Muntz and Palacios [3] showed that the product-form solution in (7) becomes

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

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

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

where $N(\underline{n})$ is the total number of customers and $N_k(\underline{n})$ is the total number of chain k customers in the network for network state $\underline{n}$. Note that if all chains are closed, $a(\underline{n}) = 1$ by definition. If at least one chain is open, then for those routing chains that are closed, say chain j for example, the product-form solution given by (25) - (26) is applicable if $\gamma_j(i)$ 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 (26) for type 2 arrivals.

One way to view a closed network is that it is an open network but the routing subchain 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 $\underline{N}$ 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 [5] that if V is an irreducible

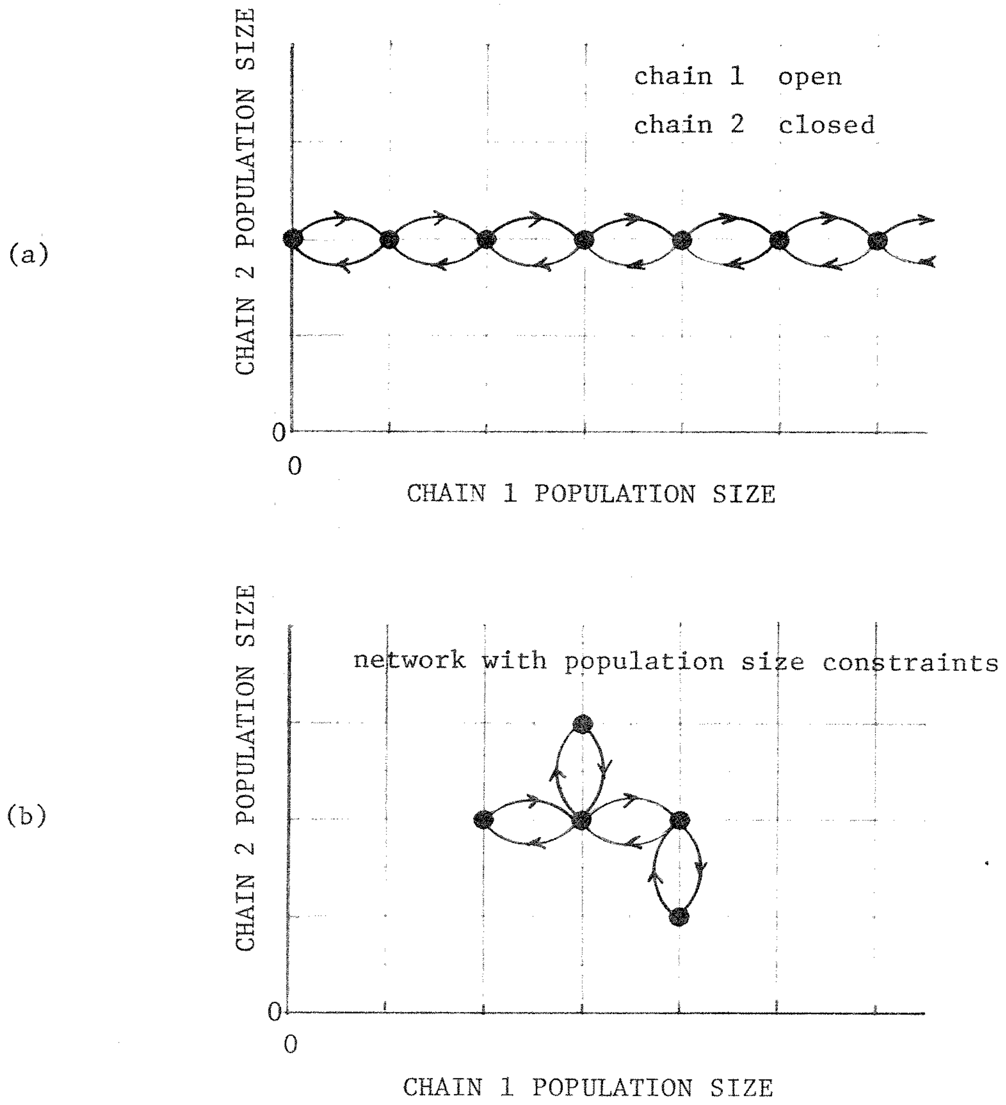

Fig. 5. Examples of two-chain networks with external arrivals and departures.

set of feasible population vectors, then a sufficient condition for the product-form solution in (25) and (26) to remain valid is: for any k, and population vectors $\underline{N}$ and $\underline{N} + \underline{1}_k$ in V, the loss mechanism is invoked for a chain k external arrival in any network state with population vector $\underline{N}$ if and only if the trigger mechanism is invoked for a chain k external departure in any network state with population vector $\underline{N} + \underline{1}_k$. (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 some special cases. The normalization constant G is given by the sum of the unnormalized product-form solution in (25) over all feasible $\underline{n}$ states for each feasible population vector in the set V.

The next theorem characterizes the behavior of the population vector $\underline{N}$ of product-form BCMP networks with external arrivals and departures and population size constraints.

Theorem (i) The population vector $\underline{N}$ is described by a continuous-time Markov chain, and (ii) the equilibrium probability distribution of $\underline{N}$ is

$$P(\underline{N}) = \frac{a(\underline{N})}{G} \quad G(\underline{\alpha},M,\underline{N}) \qquad (27)$$

where

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

and

$G(\underline{\alpha},M,\underline{N})$ is the normalization constant of an equivalent closed network with population vector $\underline{N}$ and scaling factors $\alpha_k = \lambda_{1k}$, $k = 1,2,\ldots K$, given by (23) and (24), and

$$G = \sum_{\underline{N} \text{ in } V} a(\underline{N}) \quad G(\underline{\alpha},M,\underline{N}) \tag{28}$$

We shall first consider part (i) of the theorem. Let S denote a detailed network state that is Markovian (see [3]).

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

where S_m is the state description of service center m. The equilibrium probability of S has the following form [3]

$$P(S) = \frac{\Pi^*(S)}{G}$$

$$= \frac{a(\underline{N}) \quad \Pi(S)}{G} = \frac{a(\underline{N}) \quad \Pi_1(S_1)\ \Pi_2(S_2)\ \ldots\ \Pi_M(S_M)}{G} \tag{29}$$

where $\underline{N}$ is the population vector of Markovian network state S, and $a(\underline{N})$ was defined in the theorem.

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

$$\mathcal{S} = \bigcup_{\underline{N} \text{ in } V} \mathcal{S}(\underline{N})$$

Let $\underline{N}(t)$ denote the network population vector at time t. To show part (i), it is necessary and sufficient to show that for $\underline{N}^{(1)}$ and $\underline{N}^{(2)}$ in V

$$\lim_{\Delta \to 0} \frac{1}{\Delta} \ P[\underline{N}(t) = \underline{N}^{(2)} / \ \underline{N}(t-\Delta) = \underline{N}^{(1)}\ ] = R(\underline{N}^{(1)} \to \underline{N}^{(2)}\ )$$

where $R(\underline{N}^{(1)} \rightarrow \underline{N}^{(2)})$ is a constant rate that depends only upon $\underline{N}^{(1)}$ and $\underline{N}^{(2)}$.

Transitions in V are either of the type $\underline{N} \rightarrow \underline{N} + \underline{1}_k$ or the type $\underline{N} + \underline{1}_k \rightarrow \underline{N}$. The first type of transitions corresponds to the arrival of an external customer to chain k and occurs with the rate $\gamma_k(N_k)$ or $\gamma_k(N)$. The second type of transitions corresponds to the departure of a chain k customer from the network and occurs with the following rate

$$\lim_{\Delta \to 0} \frac{1}{\Delta} P[\underline{N}(t) = \underline{N} / \underline{N}(t-\Delta) = \underline{N} + \underline{1}_k]$$

$$= \frac{a(\underline{N}+\underline{1}_k) \sum\limits_{c \text{ in } RC(k)} \; \sum\limits_{S \text{ in } \mathcal{S}(\underline{N})} \; \sum\limits_{S^{+c} \text{ in } \mathcal{S}^{+c}} \Pi(S^{+c}) R_m(S_m^{+c} \rightarrow S_m) [1 - \sum\limits_{d \text{ in } RC(K)} p_{cd}]}{a(\underline{N} + \underline{1}_k) \sum\limits_{c \text{ in } RC(k)} \; \sum\limits_{S \text{ in } \mathcal{S}(\underline{N})} \; \sum\limits_{S^{+c} \text{ in } \mathcal{S}^{+c}} \Pi(S^{+c})}$$

where S is a Markovian network state in $\mathcal{S}(\underline{N})$ and $\mathcal{S}^{+c}$ is the set of network states in $\mathcal{S}(\underline{N} + \underline{1}_k)$ that are the same as network state S but with an extra class c customer. Class c is in chain k and 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. S_m is the mth component of S. $R_m(S_m^{+c} \rightarrow S_m)$ is the transition rate from S_m^{+c} to S_m corresponding to the departure of the extra class c customer from center m, and $[1 - \sum\limits_{d \text{ in } RC(k)} p_{cd}]$ is the probability that the departing class c customer leaves the network instead of joining another service center.

After cancelling the term $a(\underline{N} + \underline{1}_k)$ in both the numerator and denominator and noting that the summation in the denominator is over all network states in $\mathcal{S}(\underline{N} + \underline{1}_k)$, the denominator is equal to the normalization constant $G(\underline{\alpha}, \underline{N} + \underline{1}_k)$ of an equivalent closed netowrk with population vector $\underline{N} + \underline{1}_k$ and scaling factors $\underline{\alpha}$. The expression in the numerator

$$\sum_{S \text{ in } \mathcal{S}(\underline{N})} \quad \sum_{S^{+c} \text{ in } \mathcal{S}^{+c}} \Pi(S^{+c}) \quad R_m(S_m^{+c} \to s_m)$$

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

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

We then have

$$\lim_{\Delta \to 0} \frac{1}{\Delta} \; P[\underline{N}(t) = \underline{N} \; / \; \underline{N}(t-\Delta) = \underline{N} + \underline{1}_k]$$

$$= \sum_{c \text{ in } RC(k)} T_c(\underline{N} + \underline{1}_k) \; [\, 1 - \sum_{d \text{ in } RC(k)} p_{cd}]$$

$$= \sum_{c \text{ in } RC(k)} \frac{G(\underline{\alpha},\underline{N})}{G(\underline{\alpha}, \; \underline{N} + \underline{1}_k)} \quad v_c[1 - \sum_{d \text{ in } RC(k)} p_{cd}]$$

$$= \frac{G(\underline{\alpha},\underline{N})}{G(\underline{\alpha},\underline{N} + \underline{1}_k)} \tag{30}$$

where the identity

$$\sum_{c \text{ in } RC(k)} v_c \; [1 - \sum_{d \text{ in } RC(k)} p_{cd}] = 1 \tag{31}$$

can be easily demonstrated using (23) and $\sum_{c \text{ in } RC(k)} q_c = 1.$

Note that λ_{mk} given by (23) - (24) can be interpreted as the mean number of visits by a chain k customer to service center m between successive visits to an "imaginary" service center acting as the source and sink of chain k customers. Recall that the throughput rate T_k of a closed chain was defined earlier to be the throughput rate of service center 1 which is arbitrarily chosen. For an open chain, it is more meaningful to define its throughput rate to be that of its imaginary source/sink service center. With the set of relative arrival rates from (23) - (24) and defining the scaling factor α_k to be λ_{1k}, the corresponding relative arrival rate to the source/sink center is unity. Hence the open chain throughput rate is

$$T_k(\underline{N} + \underline{1}_k) = \frac{G(\underline{\alpha},\underline{N})}{G(\underline{\alpha},\underline{N} + \underline{1}_k)} \qquad \text{chain } k \text{ is open} \qquad (32)$$

We have thus shown that the population vector $\underline{N}$ can be described as a continuous-time Markov chain with transition rates

$$R(\underline{N} \rightarrow \underline{N} + \underline{1}_k) = \begin{cases} \gamma_k(N) & \text{for type 1 arrivals} \\ \gamma_k(N_k) & \text{for type 2 arrivals} \end{cases}$$

and

$$R(\underline{N} + \underline{1}_k \rightarrow \underline{N}) = T_k(\underline{N} + \underline{1}_k)$$

for any pair of $\underline{N}$, $\underline{N} + \underline{1}_k$ in V.

Part (ii) of the theorem is an immediate consequence of the theorem in [5] based upon a "local balance" property of P(S). We shall provide a simpler proof of it by demonstrating a local balance property possessed by $P(\underline{N})$.

Chandy [2] first observed that the product-form solution P(S) of many queueing networks has a local balance property. This observation proved to be very useful in the discovery and characterization of the class of BCMP networks [3].

Muntz [4] found that individual service centers in BCMP networks have the M => M property, which can be explained as follows. Consider class c in service center m (viewed in isolation). Center m has the M => 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. $\Pi_m(S)$ in the product-form solution $\Pi(S)$ was found to satisfy the following sufficient condition for the M => M property:

$$\sum_{S_m^{+c} \text{ in } \mathcal{S}_m^{+c}} \frac{\Pi_m(S_m^{+c})\ R_m(S_m^{+c} \rightarrow S_m)}{\Pi_m(S_m)} = v_c \tag{33}$$

where $\mathcal{S}_m^{+c}$ is the set of center m states that are the same as state S_m but with an extra class c customer. (33) can be rewritten as

$$\Pi_m(S_m)v_c = \sum_{S_m^{+c} \text{ in } \mathcal{S}_m^{+c}} \Pi_m(S_m^{+c})R_m(S_m^{+c} \rightarrow S_m)$$

where we can interpret

(a) the LHS to be the "flow" out of state S_m due to class c arrivals, and

(b) the RHS to be the flow into state S_m due to class c departures.

The above equation 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. (A detailed treatment of local balance can be found in the work of Chandy, Howard and Towsley [13].)

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

$$\Pi(S)\ v_c = \sum_{S^{+c} \text{ in } \mathcal{S}^{+c}} \Pi(S^{+c})\ R_m(S_m^{+c} \rightarrow S_m) \tag{34}$$

which will be used 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 <u>chain local balance</u>. Consider chain k. Suppose the population vectors $\underline{N}$ and $\underline{N} + \underline{1}_k$ are feasible.

Recall the identity in (31).

$$1 = \sum_{c \text{ in } RC(k)} v_c [1 - \sum_{d \text{ in } RC(k)} p_{cd}]$$

Now replace v_c in the RHS of the above using (34) and get

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

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

$$\gamma_k(\underline{N}) = \begin{cases} \gamma_k(N) & \text{for type 1 arrivals} \\ \gamma_k(N_k) & \text{for type 2 arrivals} \end{cases} \tag{35}$$

Multiply both sides of the above equation by $\gamma_k(\underline{N})$ and rewriting $\Pi(S^{+c})/\Pi(S)$ as $\Pi^*(S^{+c})/\gamma_k(\underline{N}) \; \Pi^*(S)$, we get

$$\Pi^*(S) \; \gamma_k(\underline{N}) = \sum_{c \text{ in } RC(k)} \; \sum_{S^{+c} \text{ in } \mathcal{S}^{+c}} \Pi*(S^{+c}) R_m(S_m^{+c} \to S_m) [1 - \sum_{d \text{ in } RC(k)} p_{cd}] \tag{36}$$

which then is a local balance equation satisfied by $\Pi^*(S)$ with respect to chain k; we can interpret

(a) LHS of (36) to be the flow out of state S due to chain k arrivals;

(b) RHS of (36) to be the flow into state S due to chain k departures.

Note that (36) is applicable only if transitions between $\underline{N}$ and $\underline{N} + \underline{1}_k$ are feasible.

Let us sum (36) over S in $\mathcal{S}(\underline{N})$, and get

$$\sum_{S \text{ in } \mathcal{S}(\underline{N})} \Pi^*(S)\ \gamma_k(\underline{N}) = \sum_{S \text{ in } \mathcal{S}(\underline{N})} \ \sum_{c \text{ in } RC(k)} \ \sum_{S^{+c} \text{ in } \mathcal{S}^{+c}} \Pi^*(S^{+c})\ R_m(S_m{}^{+c} \to S_m) \times [1 - \sum_{d \text{ in } RC(k)} p_{cd}]$$

We recognize in the above equation that

$$\text{LHS} = G\ P(\underline{N})\ \gamma_k(\underline{N})$$

$$\text{RHS} = G\ T_k(\underline{N} + \underline{1}_k)\ P(\underline{N} + \underline{1}_k)$$

by recalling the definition of $P(\underline{N})$ and the definition of $T_k(\underline{N} + \underline{1}_k)$ for an open chain. Thus we have shown that if transitions are permitted between the feasible population vectors $\underline{N}$ and $\underline{N} + \underline{1}_k$ in V, then $P(\underline{N})$ and $P(\underline{N} + \underline{1}_k)$ satisfy the following local balance equation

$$P(\underline{N})\ \gamma_k(\underline{N}) = P(\underline{N} + \underline{1}_k)\ \ T_k\ (\underline{N} + \underline{1}_k) \tag{37}$$

which says that the flow out of $\underline{N}$ due to chain k arrivals is equal to the flow into $\underline{N}$ due to chain k departures.

The results in part (ii) of our theorem above are now immediate using (35), (37) together with (32) or Lemma 2. It is interesting to note that $P(\underline{N})$, and consequently P(S), is independent of feasible transitions in V imposed by the loss and trigger mechanisms. It does, however, depend upon the set V through the normalization constant G.

Corollary (M => 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 corollary is easily proved using (37) and Muntz's arguments [4].

In summary, we found that the class local balance property of $\Pi_m(S)$ of a service center in BCMP networks implies a chain local balance property of $\Pi^*(S)$, a chain local balance property of $P(\underline{N})$ and the M => M property for routing chains.

An Example

Consider a network with 2 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.

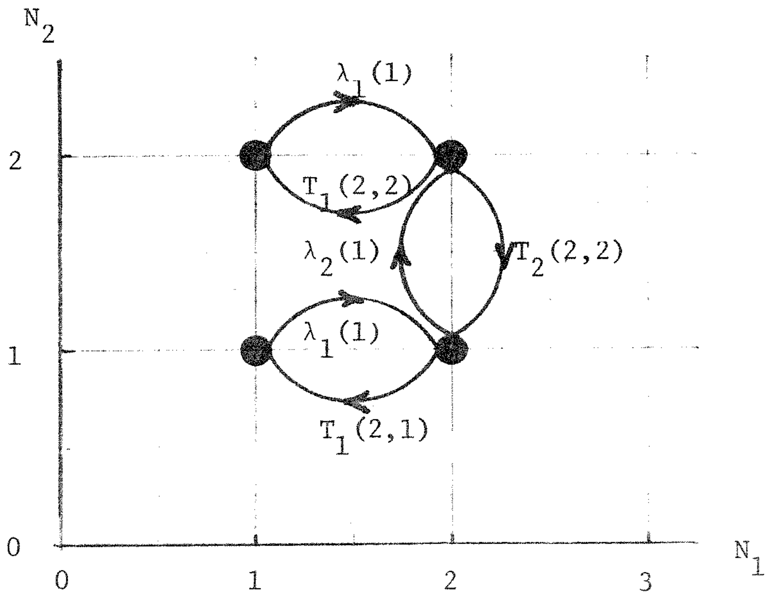

Fig. 6. An example of a two-chain network with population size constraints.

Instead of applying (27), we shall solve for $P(N_1,N_2)$ directly using the local balance equation (37), from which we get the following relationships:

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

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

and

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

Let $P(1,1) = C$ and solve for the others in terms of C.

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

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

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

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

Applying Lemma 2 to the 2 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

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

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 (28) 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, i.e.

$$\gamma_k(\underline{N}) = \gamma_k$$

then G can be found easily. First, if all chains in the network are open, then it is well-known that [3]

$$G = \prod_{m=1}^{M} \frac{1}{1 - \rho_m}$$

where

$$\rho_m = \sum_k \rho_{mk}$$

Second, if some of the chains in the network are open while the rest are closed then it was shown by Reiser and Kobayashi [7] that

$$G = G_{open} \cdot G(\underline{N})$$

where

$$G_{open} = \prod_{m=1}^{M} \frac{1}{1 - \rho_m^{o}}$$

$$\rho_m^{o} = \sum_{k\ open} \rho_{mk}$$

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

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

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

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

   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 [7]).

If the chain arrival rates $\gamma_k(\underline{N})$ depend upon the population vector and/or the network has population size constraints, then G must be evaluated from (28), repeated here

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

We mentioned earlier that V can be a very large set, possibly infinite. Note that all normalization constants $G(\underline{\alpha},M,\underline{N})$ 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 $G(\underline{\alpha},M,\underline{N})$, $\underline{N}$ in V, will fit into a given range of floating point numbers. Fortunately, 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 without affecting the accuracy of G to be evaluated!

V. CONCLUSIONS

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

$$G(\underline{\alpha},M,\underline{N}) = \prod_{k=1}^{K} (\alpha_k/\beta_k)^{N_k} \quad G(\underline{\beta},M,\underline{N})$$

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.

We also found that BCMP networks with external arrivals, departures and population size constraints can be considered as a collection of closed networks, corresponding to the set of feasible population vectors. Each feasible population vector is an aggregate state of the set of feasible states of the corresponding closed network. The behavior of the network state over the feasible population vectors is described by a continuous-time Markov chain (the time spent in each aggregate state is "memoryless"). We also showed that the class local balance property possessed by the product-form solution of BCMP networks implies a variety of interesting properties for chains.

REFERENCES

[1] Jackson, J.R., "Jobshop-like Queueing Systems", Management Science, pp. 131-142, 1963.

[2] Chandy, K. M., "The Analysis and Solutions for General Queueing Networks," Proc. 6th Annual Princeton Conf. on Information Science and Systems, 1972, pp. 224-228.

[3] Baskett, F., K. M. Chandy, R. R. Muntz, and F. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," JACM, April 1975.

[4] Muntz, R. R., "Poisson Departure Process and Queueing Networks," Proceedings of the 7th Annual Princeton Conf. on Information Sciences and System, Princeton University, Princeton, New Jersey, March 1973, pp. 435-440.

[5] Lam, S. S., "Queueing Networks with Population Size Constraints," IBM J. of Research and Development, July 1977, pp. 370-378.

[6] Buzen, J. P. "Computational Algorithms for Closed Queueing Networks with Exponential Servers," Communications of the ACM, pp. 527-531, September 1973.

[7] Reiser, M and H. Kobayashi, "Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms," IBM J. of Research and Development, May 1975.

[8] Reiser, M. and S. S. Lavenberg, "Mean Value Analysis of Closed Multichain Queueing Networks," IBM Research Report RC-7023, Yorktown Heights, NY, March 1978. To appear JACM.

[9] Chandy, K. M. and C. H. Sauer, "Computational Algorithm for Product Form Queueing Networks," presented at Performance'80, Toronto, May 1980. To appear Comm. ACM.

[10] Reiser, M., "Numerical Methods in Separable Queueing Networks," IBM Research Report RC-5842, Yorktown Heights, NY, February 1976.

[11] Chang, A. and S. S. Lavenberg, "Work Rates in Closed Queueing Networks with General Independent Servers," Operations Research, 1974, pp. 838-847.

[12] Muntz, R. R. and J. W. Wong, "Asymptotic Properties of Closed Queueing Network Models," Proceedings of the 8th Annual Princeton Conference on Information Sciences and Systems, Princeton University, Princeton, New Jersey, March 1974, pp. 348-352.

[13] Chandy, K. M., J. H. Howard and D. F. Towsley, "Product Form and Local Balance in Queueing Networks," JACM, April 1977, pp. 250-263.