AN ANALYSIS OF THE TREE CONVOLUTION ALGORITHM FOR QUEUEING NETWORKS*

Simon S. Lam and Y. Luke Lien

TR-166 February 1981

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

*This work was supported by National Science Foundation Grant No. ECS78-01803.

Abstract

An analysis of the tree convolution algorithm for the solution of product-form queueing networks is presented. A probabilistic model is used to generate networks with M service centers, K routing chains and N customers in each chain. The routes of chains are sampled from independent Bernoulli trials. The expected time and space requirements for the computation of network normalization constants and the performance measures of chain throughputs, mean queue lengths and marginal queue length distributions of service centers are derived as functions of M, K, N and the sparseness of routing chains. We assume that the same tree is employed for all networks generated by the probabilistic model. Consequently, the time and space results presented are upper bounds of the expected time and space requirements of the tree convolution algorithm in its general form.

1. INTRODUCTION

An analysis of the time and space requirements of the tree convolution algorithm [1] for the solution of product-form queueing networks is presented. A probabilistic model is employed to generate queueing networks with M service centers and K routing chains. The routes of chains are sampled from independent Bernoulli trials. We ask the following question. For a large number of networks generated, what are the average time and space requirements of the tree convolution algorithm as a function of M, K and the "sparseness" of routing chains?

It is discussed in [1] that the efficiency of the tree convolution algorithm is derived from (i) the sparseness of routing chains, and (ii) a tree planting procedure that exploits information on the routes of chains as well as any locality property present in the distribution of chains in the network.

The expected time and space requirements of the tree convolution algorithm to compute the network normalization constant and the network performance measures of chain throughputs, mean queue lengths and marginal queue length distributions of service centers are shown. We assume that the same tree is employed for all networks generated by the probabilistic model. In other words, chain routing information is not utilized to optimize the time and space requirements of the algorithm for individual networks generated.

Consequently, the results presented below are upper bounds of the expected time and space requirements of the tree convolution algorithm in its general form (i.e. with a tree planting procedure).

The probabilistic model is described in Section 2. A brief refresher of the basic ideas of the tree convolution algorithm and some key definitions is given. However, familiarity with the companion paper [1] would be most helpful for the reader to follow the development of this paper. The analysis is presented in Section 3 and Section 4. Numerical results illustrating the expected time and space requirements of the algorithm are shown in Section 5.

2. THE MODEL AND SOME DEFINITIONS

Consider a queueing network with M service centers and K routing chains. (Note that only BCMP networks with a product-form solution [2,3] can be solved by the tree convolution algorithm.) Let CENTERS(k) denote the set of service centers visited by chain k. We generate a network sample by generating the sets CENTERS(k) for $k = 1,2,\ldots,K$ according to independent Bernoulli trials. Specifically, chain k visits center m with probability P_{mk}. With probability $1-P_{mk}$, center m is not in CENTERS(k). The time and space requirements of the tree convolution algorithm depend only upon the sets CENTERS(k) and chain population sizes N_k for $k = 1,2,\ldots,K$. They do not depend upon other network parameters such as relative arrival rates and traffic intensities. (This observation assumes that the same tree is used. Also minor differences may arise if some of the centers are fixed-rate service centers and feedback filtering is employed for those convolutions involving them [1].)

For analytical tractability, we shall assume that $N_k = N$ for all k and that $P_{mk} = P$ for all m and k. It is also assumed that each chain visits at least one service center. This set of assumptions will be referred to as the <u>uniform distribution model</u> for generating networks. In this model, the probability of a chain visiting exactly i of the M service centers is

Prob [a chain visits i centers | $i \geq 1$]

$$= \frac{\binom{M}{i} P^i (1-P)^{M-i}}{1-(1-p)^M} \overset{\text{def.}}{=} b_i \qquad i = 1,2,\ldots,M. \qquad (1)$$

The mean number of centers visited by a chain is

$$\bar{n}_{\text{centers}} = \frac{MP}{1 - (1-P)^M}.$$

The ratio

$$\frac{\bar{n}_{\text{centers}}}{M} = \frac{P}{1 - (1-P)^M} \simeq P$$

may be interpreted as a measure of the sparseness of the routing chains in the network.

A balanced binary tree and postorder tree traversal are herein assumed for the tree convolution algorithm. The M service centers are fixed assigned to the leaf nodes of the tree. To further simplify the analysis, M will be assumed to be a power of 2.

We next review some key definitions and the basic tree convolution algorithm in [1]. Let SUBNET denote a subset of service centers. With respect to SUBNET, chain k is said to be fully covered if CENTERS(k) $\subseteq$ SUBNET; chain k is noncovered if the intersection of CENTERS(k) and SUBNET is null; otherwise, chain k is partially covered.

Each node in the tree has an array of values associated with it. The array p_m of a leaf node contains the (improper) marginal probabilities of queue lengths of service center m placed at that node.

Each node in the tree corresponds to a subnet of service centers that are descendants of the node in the tree. The array of a node corresponding to the set of centers SUBNET = $\{m_1, m_2, \ldots, m_s\} \subseteq \{1,2,\ldots,M\}$ is

$$g_{\text{SUBNET}} = p_{m_1} \circledast p_{m_2} \circledast \cdots \circledast p_{m_s} \tag{2}$$

where $\circledast$ denotes the convolution operation between two arrays.

The nodes in the tree are visited according to some order of tree traversal (assumed to be postorder in this paper). A branch node may be visited only after its sons have been visited. When a branch node (or the root node) is visited, its g array is computed from convolving the g arrays of its sons. Let SUBNET, SUBNET1 and SUBNET2 be sets of service centers corresponding to a branch node and its two sons respectively. Then we have

$$g_{\text{SUBNET}} = g_{\text{SUBNET1}} \circledast g_{\text{SUBNET2}}. \tag{3}$$

The root node is visited last and the network normalization constant $G(\underline{N})$ for the chain population vector $\underline{N} = (N_1, N_2, \ldots, N_k)$ is contained in the g array of the root node.

The efficiency of the tree convolution algorithm is made possible by the use of partially covered arrays in (3). Let σ_{pc} be the set of partially covered chains in SUBNET. A partially covered array for representing g_{SUBNET} has dimensionality $|\sigma_{pc}|$ and the set of index values $\{i_k = 0,1,\ldots,N_k,\ k \varepsilon \sigma_{pc}\}$. In a network of sparse routing chains, it is very likely that $|\sigma_{pc}|$ is much smaller than K for each node (subnet) in the tree. (The tree planting procedure in [1], not considered in this paper, attempts to reduce the number of partially covered chains further by exploiting information on the routes of chains.) Note that at the root node, all chains are fully covered and the g array degenerates to a single value, namely, $G(\underline{N})$.

3. EXPECTED TIME AND SPACE FOR EVALUATING THE NETWORK NORMALIZATION CONSTANT

With independent Bernoulli trials, the selection of centers visited by a chain is performed independently for all k chains.

Let us consider an individual chain and calculate the probabilities of its being fully covered, noncovered or partially covered by a subnet of x service centers. These probabilities are denoted $P_{fc}(x)$, $P_{nc}(x)$ and $P_{pc}(x)$ respectively and are given by

$$P_{fc}(x) = \sum_{i=1}^{x} b_i \text{ Prob[all i centers are in the subnet]}$$

$$= \sum_{i=1}^{x} b_i \frac{\binom{x}{i}}{\binom{M}{i}} \tag{4}$$

$$P_{nc}(x) = \sum_{i=1}^{M-x} b_i \text{ Prob[all i centers are not in the subnet]}$$

$$= \sum_{i=1}^{M-x} b_i \frac{\binom{M-x}{i}}{\binom{M}{i}} \tag{5}$$

and

$$P_{pc}(x) = 1 - P_{fc}(x) - P_{nc}(x) \tag{6}$$

where b_i in (4) and (5) is given by (1).

Since the number of partially covered chains in a subnet determines the dimensionality of its g array, it is of interest to note that

$$\text{Prob[a subnet of x centers has k partially covered chains]}$$

$$= \binom{K}{k} [P_{pc}(x)]^k [1 - P_{pc}(x)]^{K-k}. \tag{7}$$

The mean number of partially covered chains in a subnet of x centers is $KP_{pc}(x)$.

<u>Lemma 1</u>. For any integer $M \geq 2$, $P_{pc}(x)$ increases monotonically with x for $1 \leq x \leq \lfloor M/2 \rfloor$ and decreases monotonically with x for $\lceil M/2 \rceil \leq x \leq M$. $P_{pc}(x)$ equals zero at $x = M$.

The notation $\lfloor b \rfloor$ in Lemma 1 denotes the largest integer smaller than or equal to b while the notation $\lceil b \rceil$ denotes the smallest integer larger than or equal to b. Lemma 1 is proved in the Appendix.

Note that if M is an odd integer then the maximum of $P_{pc}(x)$ occurs either at $\lfloor M/2 \rfloor$ or at $\lceil M/2 \rceil$. But if M is an even integer then the maximum occurs at M/2. We shall assume M to be even from now on. In fact we shall assume it to be a power of 2.

The above lemma implies that the mean number of partially covered chains is maximized in a subnet of M/2 centers. Recall that the dimensionality of a partially covered g array for a subnet is equal to the number of partially covered chains in it. When there are j partially covered chains, the space requirement of the partially covered g array is $(N + 1)^j$.

<u>Lemma 2</u>. For a subnet of x service centers, the expected space requirement of its partially covered g array is

$$E[\text{space} | x] = [1 + NP_{pc}(x)]^K \tag{8}$$

which is maximized at $x = M/2$.

To prove Lemma 2, we have from (7)

$$E[\text{space}|x] = \sum_{j=0}^{K} (N+1)^j \binom{K}{j} [P_{pc}(x)]^j [1-P_{pc}(x)]^{K-j}$$

and (8) follows. $E[\text{space}|x]$ is maximized at $x = M/2$ since $P_{pc}(x)$ is maximized at $M/2$ from Lemma 1.

With postorder traversal of a balanced binary tree, the maximum number of g arrays needed at the same time is $(\log_2 M)+2$. This situation is illustrated in Fig. 1 for $M = 8$. Part (i) of Theorem 1 below is immediately obvious. (Note that the expectation of a sum of random variables is equal to the sum of the expectations of the random variables [4].)

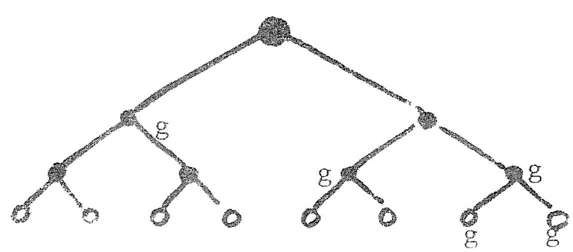

Fig. 1. Maximum number of nodes requiring g arrays at the same time.

<u>Theorem 1</u>. (i) The expected space requirement of the tree convolution algorithm to calculate the normalization constant $G(\underline{N})$ is

$$E[\text{space for } G(\underline{N})] < (2+\log_2 M)[1+NP_{pc}(M/2)]^K.$$

(ii) There exists a positive integer K' such that for $K \geq K'$

$$E[\text{space for } G(\underline{N})]$$
$$= 2[1 + NP_{pc}(M/2)]^K + 2[1 + NP_{pc}(M/4)]^K. \tag{9}$$

Part (ii) of Theorem 1 is proved by noting from Lemma 1 that there exists a positive integer K' such that for $K \geq K'$

$$\left[\frac{1 + NP_{pc}(M/2^j)}{1 + NP_{pc}(M/2^{j+1})} \right]^K > 2$$

or

$$[1 + NP_{pc}(M/2^j)]^K > 2[1 + NP_{pc}(M/2^{j+1})]^K$$
$$\text{for } 1 \leq j \leq (\log_2 M)-1.$$

This last inequality says that the expected space requirement of a branch node in the tree is larger than the sum of the expected space requirements of its two sons. To perform the sequence of convolutions corresponding to a postorder traversal of the binary tree, it is easy to see that the expected space requirement of g arrays needed at each step of the tree traversal increases monotonically, and reaches the maximum value when the convolution at the root node's right son takes place. At this time, four g arrays are needed; two for the root node's sons and two for the sons of the root node's right son. (See Fig. 2.) We have thus proved part (ii) of Theorem 1. With a little thought, it should also be obvious that (9) depends upon the use of a binary tree and does not depend upon the order of tree traversal.

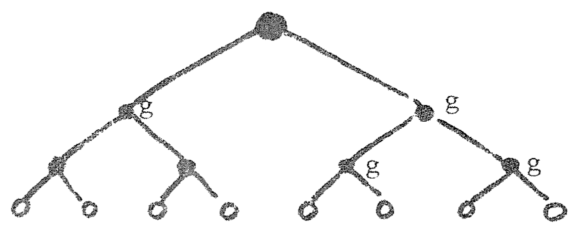

Fig. 2. Maximum expected space requirement of tree algorithm for $K \geq K'$.

We next study the expected time requirement for performing the convolution operation to merge 2 disjoint sets SUBNET1 and SUBNET2 into a set SUBNET. Consider the set of outcomes (sample space) in the sampling of centers visited by a chain according to independent Bernoulli trials. The chain is said to be <u>overlapped</u> by SUBNET1 and SUBNET2 if it is partially covered by each. First, partition the sample space of the chain outcomes into 4 events E_0, E_1, E_2 and E_3 depending upon whether the chain is partially covered by SUBNET and whether it is overlapped by SUBNET1 and SUBNET2 (see Table 1). Event E_0 is further partitioned into events E_4 and E_5 where E_4 is the set of outcomes of the chain being fully covered by either SUBNET1 or SUBNET2 and E_5 is the set of outcomes of the chain being non-covered by SUBNET. Note that the union $E_1 \cup E_3$ consists of the outcomes of the chain being partially covered by SUBNET while the union $E_2 \cup E_4$ consists of the outcomes of the chain being fully covered by SUBNET.

	not partially covered by SUBNET	partially covered by SUBNET
not overlapped	$E_0 = E_4 \cup E_5$	E_1
overlapped	E_2	E_3

Table 1. Partitioning of a chain's sample space into 5 events.

The number of centers visited by a chain is a random variable with value $i = 1,2,\ldots,M$. Suppose SUBNET1 and SUBNET2 have x centers each. The numbers of centers in SUBNET1 and SUBNET2 visited by the chain are random variables with values i_1 and i_2 respectively over the range $\{0,1,\ldots,x\}$. The five events in Table 1 are given by:

$$E_1 = (\{i_1=0\} \cup \{i_2=0\}) \cap \{1 \le i_1+i_2 \le x\} \cap \{1 \le i-i_1-i_2 \le M-2x\};$$

$$E_2 = \{1 \le i_1 \le x\} \cap \{1 \le i_2 \le x\} \cap \{i=i_1+i_2\};$$

$$E_3 = \{1 \le i_1 \le x\} \cap \{1 \le i_2 \le x\} \cap \{1 \le i-i_1-i_2 \le M-2x\};$$

$$E_4 = (\{i_1=0\} \cup \{i_2=0\}) \cap \{i=i_1+i_2\} \cap \{1 \le i \le x\};$$

$$E_5 = \{i_1=0\} \cap \{i_2=0\} \cap \{1 \le i \le M-2x\}. \tag{10}$$

The union of the five events is

$$E = \{1 \le i \le M\} \cap \{0 \le i_1 \le x\} \cap \{0 \le i_2 \le x\} \cap$$
$$\{0 \le i-i_1-i_2 \le M-2x\}$$

which is the universal event with probability 1.

The conditional probability of the event that the chain visits i_1 centers in SUBNET1 and i_2 centers in SUBNET2 given that it visits i centers altogether is

$$h(i_1,i_2|i,x) = \begin{cases} \dfrac{\binom{x}{i_1}\binom{x}{i_2}\binom{M-2x}{i-i_1-i_2}}{\binom{M}{i}} & (i,i_1,i_2)\varepsilon E \\ 0 & \text{otherwise} \end{cases}$$

We can now calculate the probability of each of the events by

$$\begin{aligned} f_y(x) &\overset{\text{def.}}{=} \text{Prob}[E_y] \\ &= \sum_{i=1}^{M} b_i \sum_{(i,i_1,i_2)\varepsilon E_y} h(i_1,i_2|i,x) \end{aligned}$$

$$\text{for } y = 0,1,2,3,4,5$$

where b_i is from (1). Note that we have the relationships

$$f_0(x) = f_4(x) + f_5(x)$$

$$P_{nc}(2x) = f_5(x)$$

$$P_{pc}(2x) = f_1(x) + f_3(x)$$

and

$$P_{fc}(2x) = f_2(x) + f_4(x).$$

Next define the random variable K_y for y = 1,2,3,4,5 representing the number of chains whose outcomes are in event E_y. Let $\underline{K}$ denote (K_1,K_2,K_3,K_4,K_5) and $\underline{k}$ denote (k_1,k_2,k_3,k_4,k_5). Define

$$p(\underline{k}|x) = \text{Prob}[\underline{K} = \underline{k}|x]$$

for $\underline{k}\varepsilon\sigma_K = \{\underline{k}|k_y=0,1,\ldots,K \text{ for } y=1,2,3,4,5 \text{ and } \sum_{y=1}^{5} k_Y = K\}$. Its

moment generating function is

$$p^*(\underline{z}|x) = \sum_{\underline{k}\varepsilon\sigma_K} p(\underline{k}|x) \quad z_1^{k_1} z_2^{k_2} z_3^{k_3} z_4^{k_4} z_5^{k_5} \tag{11}$$

where $\underline{z} = (z_1, z_2, z_3, z_4, z_5)$. Since the sampling of centers is performed independently for all k chains, we then have

$$p^*(\underline{z}|x) = [\sum_{y=1}^{5} f_y(x) z_y]^K \tag{12}$$

from the fact that the moment generating function of a sum of independent random variables is equal to the product of moment generating functions of the random variables [4].

<u>Lemma 3</u>. The expected time requirement to perform a convolution using partially covered arrays corresponding to the merger of two disjoint subnets each containing x service centers is

$$E[\text{time}|2x] = [1 + (1-f_0(x))\ N + f_3(x)N(N+1)/2]^K. \tag{13}$$

Lemma 3 is proved as follows. The time to perform a convolution using partially covered arrays is

$$(N+1)^{k_1 + k_2} [(N+2)(N+1)/2]^{k_3}$$

given by (8) in [1]. Hence the expected time is

$$E[\text{time}|2x] = \sum_{\underline{k}\varepsilon\sigma_K} p(\underline{k}|x)(N+1)^{k_1}(N+1)^{k_2}[(N+2)(N+1)/2]^{k_3}$$

$$= [f_0(x) + f_1(x)(N+1) + f_2(x)(N+1) + f_3(x)\{(N+2)(N+1)/2\}]^K$$

which simplifies to (13).

Let $p(k_{pc}, k_{fc}|x)$ be the probability of having k_{pc} partially covered chains and k_{fc} fully covered chains in a subnet of x service centers. The moment generating function of $p(k_{pc}, k_{fc}|x)$ is

$$P^*(\underline{z}|x) = [P_{nc}(x) + P_{pc}(x)z_{pc} + P_{fc}(x)z_{fc}]^K. \tag{14}$$

The time requirement to calculate the g array of a leaf node using (9) in [1] is $4(N+1)^{k_{pc} + k_{fc}}$. We then have the following result.

Lemma 4. The expected time to calculate the g array of a leaf node is

$$E[\text{time}|\text{leaf}] = 4[1 + NP_{pc}(1) + NP_{fc}(1)]^K. \qquad (15)$$

If the leaf node corresponds to a fixed-rate service center and feedback filtering using (A4) in [1] is used, then the time requirement is $(k_{pc} + k_{fc})(N + 1)^{k_{pc} + k_{fc}}$ and the following result is obtained.

Lemma 5. The expected time to calculate the g array of a leaf node using feedback filtering is

$$E[\text{time}|\text{leaf}] = K(N+1)[P_{pc}(1) + P_{fc}(1)][1 + NP_{pc}(1) + NP_{fc}(1)]^{K-1}. \qquad (16)$$

The theorem below immediately follows from the preceding three lemmas.

Theorem 2. The expected time requirement of the tree convolution algorithm to calculate the normalization constant $G(\underline{N})$ is

$$E[\text{time for } G(\underline{N})] = \sum_{j=1}^{\log_2 M} \frac{M}{2^j} E[\text{time}|2^j] + \sum_{m=1}^{M} E[\text{time}|\text{leaf } m]. \qquad (17)$$

4. EXPECTED TIME AND SPACE FOR EVALUATING NETWORK PERFORMANCE MEASURES

We shall consider the computation of $G(\underline{N} - \underline{1}_k)$ for evaluating chain throughputs, $G_{m+}(\underline{N} - \underline{1}_k)$ for evaluating mean queue lengths of fixed-rate centers and G_{m-} for obtaining the marginal distribution of queue lengths of a service center. The methods are discussed in [1].

Chain throughputs (Method 1)

To calculate the throughput of chain k we need $G(\underline{N} - \underline{1}_k)$ in addition to $G(\underline{N})$. Suppose the entire tree of g arrays from the computation of $G(\underline{N})$ is stored. The expected space requirement of the tree is

$$E[\text{space for entire tree}] = \sum_{j=0}^{\log_2 M} \frac{M}{2^j} [1 + NP_{pc}(2^j)]^K. \quad (18)$$

Define

$$P^*_{fc}(2x) = \text{Prob[chain k is fully covered by a node of 2x centers and not fully covered by either of its two sons]}$$

$$= \sum_{i=1}^{2x} B_i \binom{2x}{i} - 2 \sum_{i=1}^{x} B_i \binom{x}{i} \qquad \text{for } x = 1,2,\ldots,M/2$$

where

$$B_i = \frac{P^i(1-P)^{M-i}}{1 - (1-P)^M}.$$

Define

$$\gamma_j = \text{Prob[chain k is fully covered at level j and not fully covered at level j-1]}$$

$$= \begin{cases} \frac{M}{2^j} P^*_{fc}(2^j) & j = 1,2,\ldots,\log_2 M \\ P_{fc}(1)\, M & j = 0 \end{cases}$$

where level 0 denotes the leaf nodes of the tree and level $(\log_2 M)$ denotes the root node.

Suppose that a chain, say k, is fully covered for the first time at a level j node of the tree. To compute $G(\underline{N} - \underline{1}_k)$, the convolution at this node needs to be redone to obtain g array elements with the index value $i_k = N - 1$. The convolutions at nodes along the path from this node to the root node are then redone sequentially. The expected time of one such convolution is given by

$$E[\text{time}|2x,\ K-1] = [1 + (1-f_0(x))N + f_3(x)N(N+1)/2]^{K-1}.$$

If chain k is fully covered at a leaf node, then recomputing the g array of the leaf node for $i_k = N - 1$ using (15) in [1] requires only 4 time units (2 multiplications and 2 divisions). The expected time to compute $G(\underline{N} - \underline{1}_k)$ given that the tree of g arrays from computing $G(\underline{N})$ has been saved is

$$\begin{aligned} E[\text{time for } G(\underline{N} - \underline{1}_k)|\text{tree}] &= 4\gamma_0 + N \sum_{j=1}^{\log_2 M} \gamma_j \, E[\text{time}|2^j,\ K-1] \\ &\quad + \sum_{j=0}^{\log_2 M} \gamma_j \sum_{i=j+1}^{\log_2 M} E[\text{time}|2^i,\ K-1] \\ &= \sum_{i=1}^{\log_2 M} \gamma_i^* \, E[\text{time}|2^i,\ K-1] + 4\gamma_0 \qquad (19) \end{aligned}$$

where

$$\gamma_i^* = \sum_{j=0}^{i-1} \gamma_j + N\gamma_i .$$

Work space for two partially covered arrays is needed for the sequence of convolutions. We summarize the above results in the following theorem.

Theorem 3. Given the tree of g arrays from the computation of $G(\underline{N})$, the expected time and work space needed by the tree convolution algorithm to calculate $G(\underline{N} - \underline{1}_k)$ for all k are:

(i) $E[\text{time for } G(\underline{N} - \underline{1}_k) \text{ for all } k|\text{tree}] = K\,E[\text{time for } G(\underline{N} - \underline{1}_k)|\text{tree}]$;

(ii) $E[\text{work space for } G(\underline{N} - \underline{1}_k) \text{ for all } k|\text{tree}]$

$$= E[\text{space}|(M/2)] + E[\text{space}|(M/4)].$$

Suppose that some g arrays from the computation of $G(\underline{N})$ are not saved and stored for the subsequent computation of $G(\underline{N} - \underline{1}_k)$. Specifically, if g arrays at levels 0 to j* of the tree are not saved, then the tradeoff between expected time and expected space is as follows:

$$\text{expected space reduction} = \sum_{j=0}^{j^*} \frac{M}{2^j}\,[1 + NP_{pc}(2^j)]^K; \tag{20}$$

expected additional time

$$= K\,\left(\sum_{j=0}^{j^*} 2^{j^*+1-j}\,E[\text{time}|2^j] - \sum_{j=1}^{j^*} \bar{\gamma}_j\,E[\text{time}|2^j]\right) \tag{21}$$

where $E[\text{time}|2^0] = E[\text{time}|\text{leaf}]$ and $\bar{\gamma}_j = \sum_{i=0}^{j} \gamma_i$.

<u>Chain throughputs (Method 2)</u>

$G(\underline{N})$ and $G(\underline{N} - \underline{1}_k)$ for all k can be computed together in the same tree traversal. In this case if a node has k_{fc} fully covered chains then $k_{fc} + 1$ partially covered arrays need to be computed and stored at this node. Given also that the node has k_{pc} partially covered chains, the space requirement for the arrays is $(1 + k_{fc}) \cdot (N + 1)^{k_{pc}}$, which together with (14) yield the following result.

Lemma 6. The expected space requirement of the partially covered arrays for a subnet of x centers needed to compute $G(\underline{N} - \underline{1}_k)$ for all k is

$$E[\text{space}|x, \text{all } k] = E[\text{space}|x] + K\, P_{fc}(x)[1 + N\, P_{pc}(x)]^{K-1}.$$

Note that $P_{fc}(x)$ is equal to 1 and $P_{pc}(x)$ is equal to zero at $x = M$. Hence, the expected space is equal to $K + 1$ at the root node (i.e., one location for each of $G(\underline{N})$ and $G(\underline{N} - \underline{1}_k)$ for $k = 1,2,\ldots,K$). Since $P_{pc}(x)$ is maximized at $x = M/2$ while $P_{fc}(x)$ is a monotonically increasing function, we have the following theorem that is analogous to Theorem 1.

Theorem 4. (i) The expected space requirement of the tree convolution algorithm to compute $G(\underline{N})$ and $G(\underline{N} - \underline{1}_k)$ for all k is

$$E[\text{space for } G(\underline{N}) \text{ and } G(\underline{N} - \underline{1}_k) \text{ for all } k] < (2 + \log_2 M) \cdot \max\{E[\text{space}|M/2, \text{all } k], K+1\}.$$

(ii) There exists a positive integer K' such that for $K \geq K'$

$$E[\text{space for } G(\underline{N}) \text{ and } G(\underline{N} - \underline{1}_k) \text{ for all } k] = 2E[\text{space}|M/2, \text{all } k] + \max\{K+1, 2E[\text{space}|M/4, \text{all } k]\}.$$

We next determine the expected time requirement for computing $G(\underline{N})$ and $G(\underline{N} - \underline{1}_k)$ for all k in the same tree traversal. Suppose that k_y chains have outcomes in event E_y for $1 \leq y \leq 5$. The time requirement of convolutions performed to compute the $(1+k_2+k_4)$

arrays for the node is

$$[k_2 \frac{N}{N+1} + k_4 + 1](N+1)^{k_2} (N+1)^{k_1} [(N+2)(N+1)/2]^{k_3}$$

which results from a simple extension of (8) in [1] to include the time requirement of the additional σ arrays. The expected time requirement is then given by the following expression

$$N \frac{\partial}{\partial z_2} \; p^*(\underline{z}|x) + \frac{\partial}{\partial z_4} z_4 p^*(\underline{z}|x)$$

evaluated at $z_1 = z_2 = N+1$, $z_3 = (N+2)(N+1)/2$ and $z_4 = z_5 = 1$, which yields the following lemma.

Lemma 7. To compute $G(\underline{N})$ and $G(\underline{N} - \underline{1}_k)$ for all k in the same tree traversal, the expected time requirement at a node of 2x centers is

$$E[\text{time}|2x, \text{all } k]$$
$$= E[\text{time}|2x] + K[f_2(x)N + f_4(x)][1+(1-f_0(x))N+f_3(x)N(N+1)/2]^{K-1}.$$

The following theorem can now be stated.

Theorem 5. The expected time requirement of the tree convolution algorithm to calculate $G(\underline{N})$ and $G(\underline{N} - \underline{1}_k)$ for all k is

$$E[\text{time for } G(\underline{N}) \text{ and } G(\underline{N} - \underline{1}_k) \text{ for all } k]$$
$$= \sum_{i=1}^{\log_2 M} \frac{M}{2^j} E[\text{time}|2^j, \text{all } k]$$
$$+ M\{E[\text{time}|\text{leaf}] + 4K(P_{pc}(1) + P_{fc}(1))\}.$$

<u>Marginal distribution of queue lengths at a service center</u>

To get the marginal distribution of queue lengths at center m, we need $G_{m-}(\underline{N}-\underline{n}_m)$ for $\underline{0} \leq \underline{n}_m \leq \underline{N}$. Given that the tree of q arrays from computing $G(\underline{N})$ is stored, the array G_{m-} is obtained by first deleting center m from the tree and then redoing the convolutions along the path from center m to the root node. Note that chains that are partially covered by center m are partially covered at the root node (since center m has been deleted).

Consider the merger of SUBNET1-{m} and SUBNET2 to form SUBNET-{m}. There are x-1 centers in SUBNET1-{m}, x centers in SUBNET2 and 2x-1 centers in SUBNET-{m}. Given a chain partially covered by {m} or noncovered by {m} the conditional sample space E of chain outcomes and events E_1, E_2, E_3, E_4 and E_5 defined in Table 1 are specified in the Appendix. The conditional probabilities of these events, $f_y^a(x)$ for a noncovered chain and $f_y^b(x)$ for a partially covered chain where $1 \leq y \leq 5$, as well as results in the following lemmas are derived in the Appendix.

<u>Lemma 8</u>. The expected time to perform a convolution corresponding to the merger of a subnet of x-1 centers and a disjoint subnet of x centers in the evaluation of G_{m-} is

$$E[\text{time}\,|\,2x,m-]$$
$$= \{1 + [1 - \bar{f}_0(x)]N + \bar{f}_3(x)\ N\ (N+1)/2\}^K$$

where

$$\bar{f}_0(x) = P_{nc}(1) f_0^a(x) + P_{pc}(1) f_0^b(x) + P_{fc}(1)$$

and

$$\bar{f}_3(x) = P_{nc}(1) f_3^a(x) + P_{pc}(1) f_3^b(x).$$

Lemma 9. The expected space requirement of the partially covered array of a subnet of 2x-1 centers (where center m has been deleted) is

$$E[\text{space}|2x,m-] = [1 + N\,\bar{P}_{pc}(2x)]^K$$

where

$$\bar{P}_{pc}(2x) = P_{nc}(1)[f_1^a(x) + f_3^a(x)] + P_{pc}(1)[f_1^b(x) + f_3^b(x)]$$

which is maximized at $2x=M/2$ or $2x=M$.

The following theorem is now immediately obvious.

Theorem 6. Given the tree of g arrays from the computation of $G(\underline{N})$, the expected time and work space of the tree convolution algorithm to compute the array of G_{m-} for all m are:

(i) $E[\text{time for } G_{m-} \text{ for all } m|\text{tree}]$

$$= M \sum_{j=2}^{\log_2 M} E[\text{time}|2^j,m-];$$

(ii) $E[\text{work space for } G_{m-} \text{ for all } m|\text{tree}]$

$$= E[\text{space}|M/2,m-] + \max\{E[\text{space}|M,m-],\ E[\text{space}|M/4,m-]\}.$$

The tradeoff between time and space is as follows. Suppose g arrays at levels 0 to j* from the computation of $G(\underline{N})$ are not saved. Then we have:

$$\text{expected space reduction} = \sum_{j=0}^{j^*} \frac{M}{2^j} [1 + NP_{pc}(2^j)]^K; \qquad (20)$$

$$\text{expected additional time} = M\sum_{j=0}^{j^*} \sum_{i=0}^{j} 2^{j-i}\, E[\text{time}|2^i] \qquad (22)$$

where $E[\text{time}|2^0] = E[\text{time}|\text{leaf}]$.

Mean queue lengths for a fixed-rate service center

We need $G_{m+}(\underline{N} - \underline{1}_k)$ in addition to $G(\underline{N})$ for center m and all partially covered chains in center m. First assume that the tree of g arrays from the computation of $G(\underline{N})$ is saved. To get $G_{m+}(\underline{N} - \underline{1}_k)$, the g array at the leaf node for center m is recomputed by performing a convolution between the original g array and itself for the population vector $\underline{N} - \underline{1}_k$. If feedback filtering is used then the expected time for this convolution is given by (16). (Actually, a few extra additions are required. However the number of multiplications is the same.) Next, convolutions along the path from center m to the root node of the tree are redone for the population vector $\underline{N} - \underline{1}_k$. Suppose that multiple g arrays are computed and stored at each node in the path so that $G_{m+}(\underline{N} - \underline{1}_k)$ for all partially covered chains in center m are computed together in the same traversal.

Consider a node in the path and the events defined in Table 1. The conditional probabilities of these events, $f_y^a(x)$ for a chain noncovered by $\{m\}$ and $f_y^c(x)$ for a chain partially covered by $\{m\}$ where $1 \leq y \leq 5$, as well as results in the following lemmas are derived in the Appendix.

Lemma 10. The expected time to perform convolutions at a node of 2x centers in the evaluation of $G_{m+}(\underline{N} - \underline{1}_k)$ for all partially covered chains in center m is

$$E[\text{time}|2x,m+]$$
$$= E[\text{time}|2x] + KP_{pc}(1)[f_2^C(x)N+f_4^C(x)]F^{K-1}$$

where

$$F = 1 + [1-f_0(x)]N + f_3(x)N(N+1)/2.$$

Lemma 11. The expected space requirement of the partially covered arrays for a subnet of 2x centers in the evaluation of $G_{m+}(\underline{N} - \underline{1}_k)$ for all partially covered chains in center m is

$$E[\text{space}|2x,m+]$$
$$= E[\text{space}|2x] + KP_{pc}(1)[f_2^C(x) + f_4^C(x)][1 + NP_{pc}(2x)]^{K-1}$$

which is maximized at either $2x = M/2$ or $2x = M$.

We then have

$$E[\text{time for } G_{m+}(\underline{N} - \underline{1}_k) \text{ for all } k|\text{tree}]$$
$$= E[\text{time}|\text{leaf}] + \sum_{j=1}^{\log_2 M} E[\text{time}|2^j,\ m+] - E[\text{time}|M]$$

where E[time|leaf] is given by (16). The following theorem is easily proved.

Theorem 7. Given the tree of q arrays from the computation of $G(\underline{N})$, the expected time and work space of the tree convolution algorithm to calculate $G_{m+}(\underline{N} + \underline{1}_k)$ for all m and all partially covered chains in center m are:

(i) $E[\text{time for } G_{m+}(\underline{N} - \underline{1}_k) \text{ for all k and m}|\text{tree}]$
$$= M\, E[\text{time for } G_{m+}(\underline{N} - \underline{1}_k) \text{ for all } k|\text{tree}];$$

(ii) $E[\text{work space for } G_{m+}(\underline{N} - \underline{1}_k) \text{ for all k and m}|\text{tree}]$
$$= E[\text{space}|M/2,\ m+] + \max\{1+KP_{pc}(1),\ E[\text{space}|M/4,\ m+]\}.$$

The tradeoff between time and space is as follows. Suppose g arrays in levels 0 to j* from the computation of G(N) are not saved. Then, we have:

$$\text{expected space reduction} = \sum_{j=0}^{j^*} \frac{M}{2^j} [1 + NP_{pc}(2^j)]^K; \qquad (20)$$

expected additional time

$$= M \left(\sum_{j=0}^{j^*} 2^{j^*+1-j} E[\text{time}|2^j] - \sum_{i=1}^{j^*} E[\text{time}|2^j] \right) \qquad (23)$$

where $E[\text{time}|2^0] = E[\text{time}|\text{leaf}]$.

5. NUMERICAL RESULTS

The expected time and space requirements of the uniform distribution model are illustrated below with numerical results for $M = 64$, $K = 16$ and $N = 3$.

In Table 2, we show the probabilities $P_{pc}(2^j)$, $f_0(2^{j-1})$, $f_3(2^{j-1})$ and γ_j as well as the expected requirements $E[\text{time}|2^j]$ and $E[\text{space}|2^j]$ at different tree levels for $P = 0.05$.

In Table 3, the expected time and space requirements needed to compute $G(\underline{N})$ are shown for values of P from 0.03 to 0.12 where P is a measure of the sparseness of routing chains in the network. (We have assumed that $K \geq K'$.)

In Table 4, the expected time and space requirements to compute $G(\underline{N})$ as well as related quantities needed for the evaluation of network performance measures are shown for $P = 0.05$.

In Table 5, the space-time tradeoffs in the evaluation of network performance measures for $P = 0.05$ are shown.

For comparison, the time and space requirements to compute the normalization constant $G(\underline{N})$ of a network using the (sequential) convolution algorithm are shown in Table 6 (see [5]). Two cases are considered: (i) the general case of a network of queue-dependent service centers; and (ii) the special case of a network of fixed-rate service centers. Note that the tree convolution algorithm applies to the general case of a network of queue-dependent servers.

Also, as discussed below, the time and space requirements in Tables 2-5 are actually loose upper bounds of the expected time and space requirements of the tree convolution algorithm in its general form.

j	$P_{pc}(2^j)$	$f_0(2^{j-1})$	$f_3(2^{j-1})$	$E[space\|2^j]$	$E[time\|2^j]$	γ_j
0	4.99E-02	-	-	9.32	4.06E+01	1.31E-01
1	9.71E-02	9.03E-01	2.49E-03	5.97E+01	7.21E+01	3.46E-03
2	1.84E-01	8.16E-01	9.42E-03	1.13E+03	2.03E+03	7.28E-03
3	3.30E-01	6.68E-01	3.37E-02	6.04E+04	2.97E+05	1.62E-02
4	5.32E-01	4.58E-01	1.08E-01	4.26E+06	1.73E+08	4.01E-02
5	6.75E-01	2.61E-01	2.63E-01	4.95E+07	7.71E+10	1.26E-01
6	0	3.25E-01	0	1	4.95E+07	6.75E-01

Table 2. Probabilities and expected space-time requirements at different tree levels for $P = 0.05$.

P	E[time for G($\underline{N}$)]	E[space for G($\underline{N}$)]
0.03	2.47E+08	1.98E+06
0.04	8.97E+09	1.96E+07
0.05	1.55E+11	1.08E+08
0.06	1.50E+12	3.78E+08
0.07	9.14E+12	9.55E+08
0.08	3.90E+13	1.89E+09
0.09	1.25E+14	3.12E+09
0.10	3.20E+14	4.53E+09
0.11	6.85E+14	5.99E+09
0.12	1.27E+15	7.40E+09

Table 3. Expected space and time requirements for different values of sparseness of routing chains.

Computation of	time	work space	tree space	total space
$G(\underline{N})$	1.55E+11	1.08E+08	0	1.08E+08
$G(\underline{N}-\underline{1}_k)$ for all k	1.49E+11	5.38E+07	1.17E+08	1.70E+08
$G(\underline{N})$ and $G(\underline{N}-\underline{1}_k)$ for all k	3.04E+11	1.95E+08	0	1.95E+08
G_{m-} for all m	3.97E+12	5.20E+07	1.17E+08	1.69E+08
$G_{m+}(\underline{N}-\underline{1}_k)$ for all m and k	5.27E+12	5.59E+07	1.17E+08	1.73E+08

Table 4. A comparison of expected space and time requirements to compute normalization constants and performance measures for P = 0.05.

j^*	space reduction	additional time		
		$G(\underline{N}-\underline{1}_k)$ for all k	G_{m-} for all m	$G_{m+}(\underline{N}-\underline{1}_k)$ for all m, k
0	5.96E+02	1.30E+03	2.60E+03	5.20E+03
1	2.51E+03	4.75E+03	1.24E+04	1.50E+04
2	2.06E+04	6.99E+04	1.62E+05	1.64E+05
3	5.04E+05	8.90E+06	1.95E+07	1.95E+07
4	1.76E+07	5.00E+09	1.11E+10	1.11E+10
5	1.17E+08	2.08E+12	4.97E+12	4.97E+12

Table 5. Space-time tradeoffs if g arrays in levels 0 to j^* of tree are not saved ($P = 0.05$).

	time	space
network of queue-dependent service centers	$M[(N+1)(N+2)/2]^K$ $= 4.19E+22$	$2(N+1)^K$ $= 8.59E+09$
network of fixed-rate service centers	$MK(N+1)^K$ $= 4.40E+12$	$(N+1)^K$ $= 4.30E+09$

Table 6. Time and space requirements of the (sequential) convolution algorithm.

6. CONCLUSIONS

We have presented an analysis of the tree convolution algorithm for the solution of product-form queueing networks [1]. The analysis is based upon two assumptions. First, networks are generated by the uniform distribution model. Second, a balanced binary tree with postorder tree traversal is used for all networks. The expected time and space requirements for the computation of normalization constants and the network performance measures of chain throughputs, mean queue lengths and marginal distributions of queue lengths are derived as functions of the parameters M, K, N and the sparseness of routing chains.

Since the same tree is used for all networks, the expected time and space requirements shown herein are upper bounds of the expected time and space requirements of the tree convolution algorithm in its general form, i.e. one that employs a tree planting procedure [1] to reduce the space and time computational requirements of individual networks with the use of different trees (balanced or unbalanced). It has been our experience with tree planting procedures [1, 6] that the time and space results shown in Tables 3 and 4 are extremely loose upper bounds.

Another reason for the bounds to be loose is that we assumed $P_{mk} = P$ for all m and k in the uniform distribution model. Additional space and time reductions are possible for networks with a strong locality property (i.e. the routes of chains are clustered in certain parts of such networks) that facilitates the tree planting procedure's efforts to reduce the numbers of partially covered chains at tree nodes.

Results in Table 2 indicate that the space and time of the tree algorithm are dominated by the space and time of high-level nodes in the tree (nodes near the root). In practice, however, since these nodes correspond to large subsets of service centers, very large reductions in space and time can be obtained at these nodes. Consequently, the tree is typically not as "top-heavy" as indicated by the results of Table 2. This observation will affect the space-time tradeoffs illustrated in Table 5; the space and time requirements at low levels of the tree will become more significant (as a fraction of the total requirements) when a tree planting procedure is used than not. The above observation also has the following implications.

First, with a tree planting procedure, the ratio of work space to tree space is smaller than that of the results in Table 4.

Second, note in Table 2 that without a tree planting procedure, $\gamma_6 = 0.675$. With a tree planting procedure, more chains will become fully covered at low-level nodes of the tree. As a result, the time and space requirements of network performance measures will become relatively larger than those of $G(\underline{N})$ shown in Table 4. In particular, the time and space requirements of $G_{\underline{m}-}$ will become relatively the largest.

APPENDIX

Proof of Lemma 1

Consider the difference $\Delta(x) = P_{pc}(x+1) - P_{pc}(x)$ for $x = 1,2,\ldots,M-1$. Define

$$B_i = b_i / \binom{M}{i} = \frac{P^i(1-P)^{M-i}}{1-(1-P)^M}$$

We then have

$$\begin{aligned}\Delta(x) &= \sum_{i=1}^{M-x} B_i \binom{M-x}{i} - \sum_{i=1}^{M-x-1} B_i \binom{M-x-1}{i} \\ &\quad + \sum_{i=1}^{x} B_i \binom{x}{i} - \sum_{i=1}^{x+1} B_i \binom{x+1}{i} \\ &= \sum_{i=1}^{M-x-1} B_i \binom{M-x-1}{i-1} - \sum_{i=1}^{x} B_i \binom{x}{i-1} + B_{M-x} - B_{x+1}\end{aligned}$$

Case 1 $\quad 1 \leq x < \lfloor M/2 \rfloor$

In this case, we have the inequality $x<M-x-1$ and thus

$$\begin{aligned}\Delta(x) = \sum_{i=1}^{x} B_i\left[\binom{M-x-1}{i-1} - \binom{x}{i-1}\right] + \sum_{i=x+2}^{M-x-1} B_i \binom{M-x-1}{i-1} \\ + B_{M-x} + B_{x+1}\left[\binom{M-x-1}{x} - 1\right]\end{aligned}$$

The inequality $x<M-x-1$ also implies that

$$\binom{M-x-1}{i-1} - \binom{x}{i-1} > 0 \quad \text{and} \quad \binom{M-x-1}{x} > 1.$$

Hence, we have shown that

$$\Delta(x) > 0 \qquad \text{for } 1 \leq x < \lfloor M/2 \rfloor$$

Case 2 $\lceil M/2 \rceil \leq x < M$

In this case, we have the inequality $x > M-x-1$ and

$$\Delta(x) = \sum_{i=1}^{M-x-1} B_i \left[\binom{M-x-1}{i-1} - \binom{x}{i-1} \right] - \sum_{i=M-x+1}^{x} B_i \binom{x}{i-1}$$

$$+ B_{M-x}\left[1 - \binom{x}{M-x-1}\right] - B_{x+1}.$$

The inequality $x>M-x-1$ also implies that

$$\binom{M-x-1}{i-1} - \binom{x}{i-1} < 0 \quad \text{and } 1-\binom{x}{M-x-1} < 0.$$

Hence, we have shown that

$$\Delta(x) < 0 \quad \text{for } \lceil M/2 \rceil \leq x < M.$$

The lemma is thus proved.

Corollary. As x increases from 1 to M, $P_{fc}(x)$ increases monotonically to one and $P_{nc}(x)$ decreases monotonically to zero.

Proof of Lemma 8 and Lemma 9

Consider a chain that is either noncovered or partially covered by $\{m\}$. Let i, i_1 and i_2 denote the number of centers in SUBNET-$\{m\}$, SUBNET1-$\{m\}$ and SUBNET2 respectively visited by the chain. For a chain that is noncovered by $\{m\}$, we must have $i\geq 1$. Also, for a chain that is partially covered by $\{m\}$, we must have $i\geq 1$; otherwise, the chain would be fully covered by $\{m\}$. The conditional sample space of chain outcomes in either case is

$$E = \{0\leq i_1\leq x-1\}\cap\{0\leq i_2\leq x\}\cap\{1\leq i\leq M-1\}\cap\{0\leq i-i_1-i_2\leq M-2x\} \tag{A1}$$

Let E be partitioned into the following events:

$$e_0 = \{i_1=0\}\cap\{i_2=0\}\cap\{1\leq i\leq M-2x\};$$
$$e_1 = \{i_1=0\}\cap\{1\leq i_2\leq x\}\cap\{i=i_1+i_2\};$$
$$e_2 = \{i_2=0\}\cap\{1\leq i_1\leq x-1\}\cap\{i=i_1+i_2\};$$
$$e_3 = \{1\leq i_1\leq x-1\}\cap\{i_2=0\}\cap\{1\leq i-i_1-i_2\leq M-2x\};$$
$$e_4 = \{1\leq i_2\leq x\}\cap\{i_1=0\}\cap\{1\leq i-i_1-i_2\leq M-2x\};$$
$$e_5 = \{1\leq i_1\leq x-1\}\cap\{1\leq i_2\leq x\}\cap\{i=i_1+i_2\};$$
$$e_6 = \{1\leq i_1\leq x-1\}\cap\{1\leq i_2\leq x\}\cap\{1\leq i-i_1-i_2\leq M-2x\}.$$

Consider a chain noncovered by $\{m\}$. The five events defined in Table 1 are given by:

$$E_1 = e_3\cup e_4,$$
$$E_2 = e_5,$$
$$E_3 = e_6,$$
$$E_4 = e_1\cup e_2,$$

and

$$E_5 = e_0. \tag{A2}$$

Define the conditional probability

$$f_y^a(x) = \text{Prob}[E_y | \text{chain noncovered by } \{m\}]$$

for $1 \le y \le 5$, which is given by

$$\sum_{i=1}^{M-1} \frac{P^i(1-P)^{M-1-i}}{1-(1-P)^{M-1}} \sum_{(i,i_1,i_2)\varepsilon E_y} \binom{x-1}{i_1}\binom{x}{i_2}\binom{M-2x}{i-i_1-i_2}. \qquad \text{(A3)}$$

Consider a chain partially covered by $\{m\}$ which has been deleted from SUBNET1. The five events defined in Table 1 are given by:

$E_1 = e_1 \cup e_2 \cup e_3 \cup e_4$,

$E_2 = \{\ \}$ (null),

$E_3 = e_5 \cup e_6$,

$E_4 = \{\ \}$,

and

$E_5 = e_0$.

Define the conditional probability

$$f_y^b(x) = \text{Prob}[E_y | \text{chain partially covered by } \{m\} \text{ which is not in SUBNET1}]$$

for $1 \le y \le 5$, which is also given by (A3).

(Note that the outcomes of a chain fully covered by $\{m\}$ are always in E_5.)

The probability of having k_{nc} chains noncovered by $\{m\}$ and k_{pc} chains partially covered by $\{m\}$ has the moment generating function

$$[P_{nc}(1)z_{nc} + P_{pc}(1)z_{pc} + P_{fc}(1)]^K.$$

Given k_{nc} and k_{pc}, the conditional moment generating function of $p(\underline{k}|x)$ is

$$[\sum_{y=1}^{5} f_y^a(x) z_y]^{k_{nc}} [\sum_{y=1}^{5} f_y^b(x) z_y]^{k_{pc}}.$$

Unconditioning on k_{nc} and k_{pc} we get the moment generating function of $p(\underline{k}|x)$ to be

$$p^*(\underline{z}|x) = \{P_{nc}(1)[\sum_{y=1}^{5} f_y^a(x) z_y] + P_{pc}(1)[\sum_{y=1}^{5} f_y^b(x) z_y] + P_{fc}(1)\}^K. \tag{A4}$$

As in the proof of Lemma 3, the expected time in Lemma 8 is obtained by evaluating the above moment generating function at $z_0=1$, $z_1=z_2=N+1$ and $z_3=(N+2)(N+1)/2$.

$P^*(\underline{z}|x)$ defined in (14) can be obtained from (A4) with z_{pc} replacing z_1 and z_3, z_{fc} replacing z_2 and z_4, and setting z_5 equal to 1. The result in Lemma 9 is then readily obtained in the same manner as that in Lemma 2.

To show that E[space|2x,m-] is maximized at 2x=M/2 or 2x=M, it is sufficient to note that

(i) $f_1^a(x)+f_3^a(x)$ is the same as $P_{pc}(2x-1)$ in Lemma 1 but with M-1 instead of M centers;

(ii) $f_1^b(x)+f_3^b(x) = 1 - \sum_{i=1}^{M-1-x} \frac{P^i(1-P)^{M-1-i}}{1-(1-P)^{M-1}} \binom{M-1-x}{i}$

which increases monotonically with x and is equal to one at x = M-1.

Proof of Lemma 10 and Lemma 11

Given a chain that is noncovered by $\{m\}$, both the conditional sample space E and the events E_y for $0\leq y\leq 5$ are the same as those given in (A1) and (A2). Hence, the conditional probability $f_y^a(x)$ of event E_y for $0\leq y\leq 5$ is also the same as before.

Given a chain that is partially covered by $\{m\}$, the conditional sample space E is the same as (A1) but the events $\{E_y\}$ are now given by

$$E_1 = e_0 \cup e_3,$$
$$E_2 = e_1 \cup e_5,$$
$$E_3 = e_4 \cup e_6,$$
$$E_4 = E_0 = e_2$$

and

$$E_5 = \{\quad\} \quad \text{(null)}. \tag{A5}$$

The conditional probability of event E_y is

$$f_y^c(x) = \text{Prob}[E_y | \text{chain partially covered by } \{m\}] \quad 0\leq y\leq 5$$

which can be computed using (A3).

Let k_y^c be the number of chains partially covered by $\{m\}$, that have outcomes in event E_y for $1\leq y\leq 5$; similarly define k_y^a for chains noncovered by $\{m\}$. Thus, k_y defined earlier is equal to $k_y^a + k_y^c$ for $1\leq y\leq 5$. Redefine $\underline{k}$ to be $\{k_y^a, k_y^c \text{ for } 1\leq y\leq 5\}$ and $\underline{z}$ to be $\{z_y^a, z_y^c \text{ for } 1\leq y\leq 5\}$. In a manner similar to the derivation of (A4), the moment generating function of $p(\underline{k}|x)$ is obtained to be

$$p^*(\underline{z}|x) = \{P_{nc}(1)[\sum_{y=1}^{5} f_y^a(x) z_y^a] + P_{pc}(1)[\sum_{y=1}^{5} f_y^c(x) z_y^c] + P_{nc}(1)\}^K. \tag{A6}$$

Conditioning on $\underline{k}$, the time requirement to compute the necessary partially covered g arrays at a node of 2x centers is

$$[k_2^c \frac{N}{N+1} + k_4^c + 1](N+1)^{k_2}(N+1)^{k_1}[(N+2)(N+1)/2]^{k_3}$$

which together with (A6) yield Lemma 10. We have made use of the relationships

$$f_y(x) = P_{nc}(1)\ f_y^a(x) + P_{pc}(1)\ f_y^c(x) \qquad y=1,2,3,5$$

and

$$f_4(x) = P_{nc}(1)\ f_4^a(x) + P_{pc}(1)\ f_4^c(x) + P_{nc}(1).$$

Similarly, conditioning on $\underline{k}$, the space requirement of partially covered g arrays at a node of 2x centers is

$$(1 + k_2^c + k_4^c)\ (N+1)^{k_{pc}}$$

which together with (A6) yield the expected space requirement in Lemma 11. To show that E[space|2x, m+] is maximized at 2x = M/2 or 2x = M, note that $f_2^c(x) + f_4^c(x)$ is monotonically increasing with x.

REFERENCES

[1] Lam, S. S. and Y. L. Lien, "A Tree Convolution Algorithm for the Solution of Queueing Networks," Department of Computer Sciences, University of Texas at Austin, Technical Report TR-165, January 1981.

[2] 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.

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

[4] Drake, A. W., Fundamentals of Applied Probability Theory, McGraw-Hill, New York, 1967.

[5] Zahorjan, J., "The Approximate Solution of Large Queueing Network Models," Ph.D. Thesis, available as Technical Report CSRG-122, Computer Systems Research Group, Univ. of Toronto, August 1980.

[6] Lien, Y. L., "Modeling and Analysis of Flow-controlled Computer Communication Networks," Ph.D. Thesis, Dept. of Computer Sciences, University of Texas at Austin, 1981; in preparation.