Computing Approximate Blocking Probabilities for Large Loss Networks with State-Dependent Routing

Shun-Ping Chung, Arik Kashper, and Keith W. Ross

Abstract— **We consider a reduced load approximation (also referred to as an Erlang fixed point approximation) for estimating point-to-point blocking probabilities in loss networks (e.g., circuit switched networks) with state-dependent routing. In this approximation scheme, the idle capacity distribution for each link in the network is approximated, assuming that these distributions are independent from link to link. This leads to a set of nonlinear fixed-point equations which can be solved by repeated substitutions. We examine the accuracy and the computational requirements of the approximation procedure for a particular routing scheme, namely least loaded routing. Numerical results for six-node and 36-node asymmetric networks are given. A novel reduced load approximation for multirate networks with state-dependent routing is also presented.**

I. Introduction

In this paper, we examine the accuracy and the computational requirements of a reduced load approximation applied to estimating point-to-point blocking probabilities for loss networks with state-dependent routing.

A loss network is typically modeled as a mutlidimensional Markov process, where the dimension of the process is equal to the number of routes permitted in the network. If alternative routes are present, the Markov process does not admit a product form solution, and the equilibrium state probabilities can be obtained by solving the linear equations associated with the generator of the process. This approach must be ruled out, however, since networks of practical interest can have hundreds of thousands of routes, and the number of states grows exponentially with the number of routes.

It is, therefore, of interest to develop computational procedures that accurately approximate blocking probabilities for loss networks. One such method, the reduced load approximation (also referred to as the Erlang fixed-point approximation), proposed as early as 1964 [5], [20] has enjoyed the attention of numerous researchers in recent years.

For the case of fixed routing, i.e., no alternative routes, this scheme assumes that blocking occurs independently from link to link and that the offered traffic to a given link is Poisson but thinned by blocking on other links. This leads to a set of nonlinear fixed point equations with the approximate blocking probabilities at the various links as the unknown variables. Repeated substitution is typically suggested for solving the fixed point equations. The reader is referred to [40], [23], [24], [29], [7], [16], [18], [43], [42], [13], [4] and the references therein for recent developments on the reduced load approximation for fixed routing.

The reduced load approximation can be extended in a natural manner to sequential alternative routing with trunk reservation [1], [33]. It is shown in [1] that the corresponding fixed point equations do not necessarily have a unique solution; however, it has been observed that if there is sufficient trunk reservation, then there is a unique solution. Moreover, the approximation gives blocking probabilities that are close to the exact values [33], and the computational effort is not significantly greater than that for fixed routing. (However, it has been shown by Hunt [17] that with alternative routing the approximation is *not* asymptotically correct under a limiting regime with large link capacities and large offered loads.) The approximation can also be extended to cover Dynamic Alternative Routing (DAR) [12]; see also [26], [10], [11] as well as the excellent survey paper [25] on loss networks.

Recently, telecommunication companies have begun to implement state-dependent routing schemes in circuit-switched networks by making use of common channel signaling and stored program control [19], [2], [3], [34], [28], [6]. In these schemes, routing decisions are based on the current number of idle circuits in each of the links throughout the network. For example, in Least Loaded Routing (LLR), if the call cannot be set up along the direct route, the two-link alternative route with the largest number of point-to-point free circuits is chosen. A version of LLR has recently been implemented in AT&T's long-distance domestic network [2].

Girard and Bell [15], [14] give an approximation procedure for one such dynamic routing scheme, Dynamic Call Routing (DCR). They report poor accuracy for a ten-node asymmetric network (weighted average blocking was overestimated by more than 2% for a wide range of loads). Krishnan [27] proposes an approximation procedure for a different state-dependent routing scheme; average blocking probabilities are again significantly overestimated. In both of these schemes, the offered traffic to a link is approximated as a Poisson process.

Kelly [24] gives a generalized reduced load approximation that can be adapted to essentially any dynamic routing scheme. This reduced load approximation is a natural generalization of those used for fixed routing, sequential routing, and DAR. Here

Manuscript received July 1991; revised June 1992; recommended for transfer from the IEEE TRANSACTIONS ON COMMUNICATIONS by IEEE/ACM TRANSACTIONS ON NETWORKING Editor Debasis Mitra. This research was partially supported by AT&T Bell Laboratories and NSF Grant NCR-891447.

S.-P. Chung is with the National Taiwan Institute of Technology, Taipei, Taiwan 10772.

A. Kashper is with AT&T Bell Laboratories, Holmdel, NJ 07733.

K. W. Ross is with the Department of Systems, University of Pennsylvania, Philadelphia, PA 19104. (email: eniac.seas.upenn.edu!ross)

IEEE Log Number 9206162.

the idle capacity *distribution* for each link in the network is approximated. For a given link, the idle capacity distribution is obtained by assuming that interarrival times are exponentially distributed with rate depending on the number of idle circuits in that link. This assumption enables one to use the well-known formula for the equilibrium probabilities for a birth/death process in order to approximate the idle capacity distributions. Kelly studies neither the computational effort nor the accuracy of the approximation scheme. Furthermore, Kelly does not address the problem of calculating the state-dependent arrival rates from the link occupancy distributions. Independently, Wong and Yum [41] proposed this same approximation specialized to LLR on symmetric networks. (In the case of a symmetric network, the computational effort becomes significantly reduced.)

Mitra, Gibbens, and Huang [32], [31], [30] have recently carried out an important theoretical study of this reduced load approximation applied to an aggregated version of LLR for symmetric networks. (In the case of aggregated LLR, link occupancies are grouped into aggregate states, and routing decisions are based on the aggregate states of the links.) The empirical testing in [32], [31] shows that aggregate LLR with a small number of aggregates can give approximate blocking probability that is very close to that of LLR. The asymptotic properties of the approximation scheme, applied to two-aggregate LLR, as the number of nodes becomes large is also studied in [31]. It is shown that if the offered load is below a certain critical value, then blocking goes to zero exponentially fast; however, if the offered load is above the critical value, the blocking probability converges to a positive value (depending on the link capacity and offered load). Other asymptotic regimes are studied in [30], giving rise to additional theoretical insights.

In this paper, we explore the accuracy and computational effort of the generalized reduced load approximation for state-dependent routing over asymmetric networks. In Section II, we review the reduced load approximation for general state-dependent routing schemes. In Section III, we obtain explicit expressions for the state-dependent arrival rates for the case of LLR over asymmetric networks. We then outline two implementations of repeated substitution for LLR. The first implementation requires $O(CN^4)$ operations per iteration of repeated substitution and $O(CN^2)$ memory, where C is the number of circuits in a link and N is the number of nodes. The second implementation trades off CPU time for memory—it requires $O(CN^3)$ operations per iteration and $O(CN^3)$ memory. We also introduce a cruder approximation scheme which attempts to reduce the computational effort of large values of C. In Section IV, we then compare the various approximation techniques with simulation for six-node and 36-node asymmetric networks which employ LLR. In Section V, we present a novel approximation scheme for state-dependent routing with multirate traffic.

Finally, in Section VI, we summarize our findings and identify areas for future research. In particular, we conclude that the reduced load approximation considered here is significantly more accurate than the approximation schemes proposed in [15], [14], [27]. However, if the traffic is in a certain *critical region*, the approximation considered here for LLR can underestimate blocking by a wide margin. Furthermore, the computational and memory requirements of the scheme can be important, perhaps excessive, for large networks.

II. A Reduced Load Approximation for State- Dependent Routing

We now describe an approximation method, apparently first noted by Kelly [24], which applies to a large class of routing schemes. In order to simplify the notation, we present a version of the method that is applicable to a (slightly) smaller class of routing schemes.

Consider a network with J links connected in an arbitrary topology. Denote C_j for the number of circuits in link j. At a given instant of time, some of the circuits in link j will be busy and the remainder will be idle. Let m_j denote the number of idle circuits on link j, and let $\boldsymbol{m} = (m_1, \cdots, m_J)$ denote the network state. The state space is given by $\Lambda = \{0, \cdots, C_1\} \times \cdots \times \{0, \cdots, C_J\}$.

A *route* R is a subset of links from $\{1, 2, \cdots, J\}$. In general, there can be $2^J - 1$ routes, although there is much less in practice. Denote $\mathcal{R}_j$ for the set of routes that employ link j.

In order for a call to be set up on route R, at least one circuit must be idle in each link $j \in R$. Denote the rate at which calls are *set up* on route R when the network is in state $\boldsymbol{m}$ by $\lambda_R(\boldsymbol{m})$. Clearly $\lambda_R(\boldsymbol{m})$ must satisfy

$$\lambda_R(\boldsymbol{m}) = 0 \quad \text{if} \quad m_j = 0 \quad \text{for some} \quad j \in R. \tag{1}$$

As an example, consider the case of *fixed routing* where calls arrive to route R with rate a_R, and a call is set up on route R if and only if $m_j > 0$ for all $j \in R$. Thus,

$$\lambda_R(\boldsymbol{m}) = \begin{cases} a_R & \text{if } m_j > 0 \text{ for all } \boldsymbol{j} \in R \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Expressions for $\lambda_R(\boldsymbol{m})$ for least loaded routing will be given in Section III.

Returning to general state-dependent routing schemes, let X_j be a random variable equal to the number of idle circuits on link j in equilibrium. Let $\boldsymbol{X} = (X_1, \cdots, X_J)$ and let

$$q_j(m) = P(X_j = m), \qquad m = 0, \cdots, C_j$$

be the idle capacity distributions. Throughout, the following approximation is made: the random variables $X_1, \cdots, X_J$ are mutually independent. Denote

$$q(\boldsymbol{m}) = \prod_{j=1}^{J} q_j(m_j), \qquad \boldsymbol{m} \in \Lambda \tag{3}$$

and $\boldsymbol{q} = (q(\boldsymbol{m}) : \boldsymbol{m} \in \Lambda)$ for the probability measure over Λ defined by (3).

The second approximation that is made is: When there are m idle circuits in link j, the time until the next call is set up on link j is exponentially distributed with parameter $\alpha_j(m)$, where

$$\alpha_j(m) = \sum_{R \in \mathcal{R}_j} E_{\boldsymbol{q}}[\lambda_R(\boldsymbol{X}) | X_j = m]. \tag{4}$$

In (4), $E_{\boldsymbol{q}}[\lambda_R(\boldsymbol{X})|X_j = m]$ is the expected setup rate of calls on route R when m circuits are available on link j. By adding this quantity over $R \in \mathcal{R}_j$, we obtain the total expected setup rate on link j when there are m circuits available on link j. Note that we have subscripted the expectation operator with $\boldsymbol{q}$ to emphasize the dependence on the marginal probabilities $q_j(\cdot), j = 1, \cdots, J$. Also note that (1) implies $\alpha_j(0) = 0$ for all $j = 1, \cdots, J$. We also assume that the holding times of all calls are exponentially distributed with unit mean.

Since interarrivals to links are assumed to be exponentially distributed with parameter $\alpha_j(m)$, it follows that

$$q_j(m) = \frac{C_j(C_j-1)\cdots(C_j-m+1)}{\alpha_j(1)\,\alpha_j(2)\cdots\alpha_j(m)}\, q_j(0), \qquad m = 1, \cdots, C_j \tag{5}$$

where

$$q_j(0) = \left[1 + \sum_{m=1}^{C_j} \frac{C_j(C_j-1)\cdots(C_j-m+1)}{\alpha_j(1)\,\alpha_j(2)\cdots\alpha_j(m)}\right]^{-1}. \tag{6}$$

Equations (3)–(6) lead to an iterative algorithm that produces an approximation for the idle capacity distributions:

1. Choose $\alpha_j(\cdot), j = 1, \cdots, J$, arbitrarily.
2. Determine $\boldsymbol{q}$ from (5), (6), and (3).
3. Obtain new values of $\alpha_j(\cdot), j = 1, \cdots, J$ through (4). Go to (2).

Because this scheme is a generalization of the reduced load approximation applied to sequential routing (see below), convergence is not guaranteed [1], although it will occur in many practical circumstances.

For certain dynamic routing schemes, it may be a nontrivial task to calculate the expectations in (4) (with $\boldsymbol{q}$ given). However, we shall see below and in Section III that tractable expressions are available for the expected arrival rate $\alpha_j(m)$ for many important routing schemes.

A. Examples

In order to gain some insight into the reduced load approximation (3)–(6), we consider three particular examples. First, we consider fixed routing. With $m > 0$, we have

$$\begin{aligned}\alpha_j(m) &= \sum_{R \in \mathcal{R}_j} E_{\boldsymbol{q}}[\lambda_R(\boldsymbol{X})|X_j = m] \\ &= \sum_{R \in \mathcal{R}_j} a_R P_{\boldsymbol{q}}(X_i > 0, i \in R - \{j\}) \\ &= \sum_{R \in \mathcal{R}_j} a_R \prod_{\substack{i \in R \\ i \neq j}} [1 - q_i(0)]\end{aligned}$$

where we have used (2) to obtain the second equality. Note that this is the standard approximation [40], [23] for the offered load to link j for fixed routing. Also note that only $\alpha_j(1)$ and $q_j(0), j = 1, \cdots, J$ must be calculated at each iteration of repeated substitution.

As a second example, we consider *sequential routing* for the simple three-node fully connected network. We assume that routing is done as follows. When a call requests a connection between the pair of nodes directly connected by link 1, setup is first attempted along link 1. If $m_1 = 0$, then setup is attempted in the route $\{2,3\}$. The call setup is completed in $\{2,3\}$ if and only if $m_2 > r_2$ and $m_3 > r_3$, where r_1, r_2, r_3 are given *trunk reservation thresholds*. Routing for a call that requests a connection between the other two pairs of nodes is carried out in an analogous manner. Examples of some state-dependent call setup rates for this routing scheme are given below:

$$\begin{aligned}\lambda_{\{1\}}(\boldsymbol{m}) &= a_1 1(m_1 > 0) \\ \lambda_{\{1,3\}}(\boldsymbol{m}) &= a_2 1(m_2 = 0, m_1 > r_1, m_3 > r_3) \\ \lambda_{\{1,2\}}(\boldsymbol{m}) &= a_3 1(m_3 = 0, m_1 > r_1, m_2 > r_2)\end{aligned}$$

where a_j is the exogenous arrival rate for the node pair directly connected by link j. Inserting the above equations into (4) gives

$$\alpha_1(m) = \begin{cases} 0 & m = 0 \\ a_1 & 0 < m \le r_1 \\ a_1 + a_2 q_2(0)\left[\displaystyle\sum_{n=r_3+1}^{C_3} q_3(n)\right] & \\ \qquad + a_3 q_3(0)\left[\displaystyle\sum_{n=r_2+1}^{C_2} q_2(n)\right] & r_1 < m \le C_1. \end{cases}$$

Note that this is the standard formula for the offered load to a link for sequential routing with trunk reservation (e.g., see [1], [33]).

As a third example, we consider the same three-node network with the state-dependent routing scheme that always seeks the most available route. That is, when a call requests a connection between the pair of nodes directly connected by link 1, the call is set up on link 1 if and only if $m_1 > 0$ and $m_1 \ge \min(m_2, m_3)$. If $m_1 < \min(m_2, m_3)$, then the call is set up in the route $R = \{2,3\}$. Note that $\min(m_2, m_3)$ is the number of idle point-to-point circuits on route $R = \{2,3\}$. The routing policies for calls with direct link 2 and direct link 3 are defined in a completely analogous manner. In this case, we have the following state-dependent call setup rates:

$$\begin{aligned}\lambda_{\{1\}}(\boldsymbol{m}) &= a_1 1(m_1 > 0, m_1 \ge m_2 \wedge m_3) \\ \lambda_{\{1,3\}}(\boldsymbol{m}) &= a_2 1(m_2 < m_1 \wedge m_3) \\ \lambda_{\{1,2\}}(\boldsymbol{m}) &= a_3 1(m_3 < m_1 \wedge m_2)\end{aligned}$$

where $x \wedge y := \min(x, y)$. Inserting the above equations into (4) gives for $m > 0$

$$\begin{aligned}\alpha_1(m) = a_1 P_{\boldsymbol{q}}(m \ge X_2 \wedge X_3) + a_2 P_{\boldsymbol{q}}(X_2 < m \wedge X_3) \\ + a_3 P_{\boldsymbol{q}}(X_3 < m \wedge X_2).\end{aligned}$$

III. Least Loaded Routing

In this section, we show how $\alpha_j(m)$ can be calculated for least loaded routing for a fully connected network with an arbitrary number of nodes. For notational convenience, we suppose that the trunk reservation level is the same for each link, and we denote it by r. Let N be the number of nodes so that the number of links is $J = N(N-1)/2$.

Each pair of nodes has an associated direct route $\{j\}$ and a set of $N-2$ alternative two-link routes, denoted by $\mathcal{A}_j$. Let

the routes in $\mathcal{A}_j$ be ordered in some manner. Let

$$m_R = \min\{m_i : i \in R\}$$

which is the number of free point-to-point circuits on route R. The routing algorithm operates as follows. When a call arrives, it is set up on the direct route $\{j\}$ if $m_j > 0$. Otherwise, setup is attempted on the least loaded alternative route R^*, where R^* maximizes m_R over $R \in \mathcal{A}_j$. In the case of ties, R is chosen from the tie set according to the ordering of $\mathcal{A}_j$. (Another possibility is to choose at random from tie set. It turns out that this minor change complicates the analysis significantly for asymmetric networks.) If $m_{R^*} \leq r$, then the call is blocked; otherwise it is set up on the route R^*. Let a_j be the exogenous arrival rate for the node pair directly connected by link j.

We need to introduce some additional notation in order to give an explicit expression for the expected setup rate $\alpha_j(m)$ for this routing scheme. If link j belongs to one of the routes in the ordered set $\mathcal{A}_k$, where k is some other link, let $\mathcal{A}_k^-(j) \subset \mathcal{A}_k$ be the set of routes that precede that route, and $\mathcal{A}_k^+(j) \subset \mathcal{A}_k$ be the set of routes that succeed that route. Let S_j be the set of links adjacent to link j (S_j contains $2(N-2)$ links). If links j and k are adjacent, then there is a third link that forms a triangle with links j and k. Let X_{jk} denote the number of idle circuits on this third link. Finally, let $Y_R = \min\{X_i : i \in R\}$ be the number of idle point-to-point circuits on route R (i.e., Y_R is the random variable corresponding to m_R). With this notation, we have $\alpha_j(0) = 0$,

$$\begin{aligned}\alpha_j(m) &= a_j \text{ for } 1 \leq m \leq r, \text{ and for } m > r \\ \alpha_j(m) &= a_j + \sum_{k \in S_j} a_k P(X_k = 0) \\ &\quad P(m \wedge X_{jk} > Y_R, R \in \mathcal{A}_k^-(j), \\ &\quad m \wedge X_{jk} \geq Y_R, R \in \mathcal{A}_k^+(j), X_{jk} > r). \qquad (7)\end{aligned}$$

The first term in (7) is due to the direct traffic on link j, whereas the second term is due to the indirect traffic on link j. Indirect traffic on link j results from direct traffic on any of its adjacent links $k \in S_j$ that overflows and is then carried on the alternative route containing link j. The probability that a call overflows on link k is $P(X_k = 0)$; the probability that it is then carried on the alternative route containing link j is

$$\begin{aligned}&P(m \wedge X_{jk} > Y_R, R \in \mathcal{A}_k^-(j), m \wedge X_{jk} \geq Y_R, \\ &\quad R \in \mathcal{A}_k^+(j), X_{jk} > r). \qquad (8)\end{aligned}$$

In words, (8) is the probability that the number of idle point-to-point circuits in the alternative route that includes link j is greater than the number of idle circuits in the preceding routes $R \in \mathcal{A}_k^-(j)$ and is greater than or equal to the number of idle circuits in the succeeding routes $R \in \mathcal{A}_k^+(j)$. Note that the last event in (8) reflects the requirement that in order to set up a call on an alternative route, the number of free circuits in each of its links must be greater than the trunk reservation level.

Conditioning on X_{jk} in (7) and employing the independence assumption gives for $m > r$

$$\begin{aligned}\alpha_j(m) = a_j + \sum_{k \in S_j} a_k P(X_k = 0)\big[&h(j,k,m) \\ &+ P(X_{jk} > m)\, g(j,k,m)\big] \qquad (9)\end{aligned}$$

where

$$h(j,k,m) = \sum_{l=r+1}^{m} P(X_{jk} = l)\, g(j,k,l) \qquad (10)$$

and where

$$g(j,k,l) = \prod_{R \in \mathcal{A}_k^-(j)} P(Y_R < l) \prod_{R \in \mathcal{A}_k^+(j)} P(Y_R \leq l). \qquad (11)$$

Note that

$$P(Y_R < l) = 1 - \prod_{j \in R} P(X_j \geq l). \qquad (12)$$

Thus, given $\boldsymbol{q}$, the expected setup rate $\alpha_j(m)$ can be calculated with (9)–(12). Once all the $\alpha_j(m)$'s are obtained, a new value of $\boldsymbol{q}$ can be calculated with (5). Once having converged on a $\boldsymbol{q}$, the blocking probability for the traffic between the node pair directly connected by link j is approximated by

$$L_j = q_j(0) \prod_{R \in \mathcal{A}_j} \left[1 - \prod_{i \in R} P_{\boldsymbol{q}}(X_i > r)\right]. \qquad (13)$$

A. Computational Considerations

Suppose, at a given iteration of repeated substitution, we have a current value of $\boldsymbol{q} = \{q_j(n); 0 \leq n \leq C_j, j = 1, \cdots, J\}$. How much computational effort is required to obtain a new value of $\boldsymbol{q}$ via (9)–(12)? To answer this question, let us assume for simplicity that $C_j = C$ for $j = 1, \cdots, J$. Note that $O(CN^2)$ memory is required to store $\boldsymbol{q}$. Since $\boldsymbol{q}$ must be updated at each iteration, it follows that $O(CN^2)$ is a lower bound for both memory and computational requirements. In the discussion that follows, assume that along with $\boldsymbol{q}$, the values $P(X_j \geq l), l = 0, \cdots, C, j = 1, \cdots, J$ are stored in memory.

Calculating $\boldsymbol{q}$ from $\alpha_j(\cdot), j = 1, \cdots, J$ requires $O(CN^2)$ operations. Consider the following algorithm to calculate $\alpha_j(\cdot), j = 1, \cdots, J$ from $\boldsymbol{q}$.

First Algorithm:

1. Do for $j = 1, \cdots, J$.
   2. Do for $k \in S_j$.
      3. Do for $l = r, \cdots, C$.
         4. Calculate $P(Y_R \leq l)$ via (12) for all $R \in \mathcal{A}_k^-(j) \cup \mathcal{A}_k^+(j)$.
      5. Do for $l = r, \cdots, C$.
         6. Calculate $g(j,k,l)$ via (11).
      7. Calcualte $h(j,k,m)$ for $m = r+1, \cdots, C$ recursively via (10).
   8. Do for $m = r+1, \cdots, C$.
   9. Calculate $\alpha_j(m)$ via (9).

Steps 4 and 6 each require $O(N)$ operations; therefore, Steps 3–6 require $O(CN)$ operations. And since Step 7 requires $O(C)$ operations and $|S_j| = 2(N-2)$, it follows that the Do loop in Step 2 requires $O(CN^2)$ operations. Since Step 2 is called $J = N(N-1)/2$ times, it follows that the above algorithm requires a total of $O(CN^4)$ operations. It can also be seen that the memory required by this approach is $O(CN^2)$.

In the previous algorithm, for a given l and R, the value $P(Y_R \leq l)$ will be calculated many times. The following algorithm, which also calculates $\alpha_j(m), m = r+1, \cdots, C$, removes this redundancy at the expense of additional memory.

Second Algorithm:

1. Do for $k = 1, \cdots, J$.
   2. Do for $l = r, \cdots, C$.
      3. Calculate $P(Y_R \leq l)$ via (12) for all $R \in \mathcal{A}_k$.
   4. Do for $l = r, \cdots, C$.
      5. Calculate $g(j, k, l)$ via (11) for all $j \in S_k$.
6. Do for $j = 1, \cdots, J$.
   7. Calculate $h(j, k, m)$ for $m = r+1, \cdots, C$ recursively via (10).
   8. Do for $m = r+1, \cdots, C$.
   9. Calculate $\alpha_j(m)$ via (9).

Note that, in this algorithm, each $P(Y_R \leq l)$ is calculated exactly once in the Do loop of Step 1. Also note, that for a given k and l, Step 5 can be done with $O(N)$ operations. Thus, this algorithm requires a total of $O(CN^3)$ operations; however, since all of the $g(j, k, l)$'s must now be stored, $O(CN^3)$ memory is required.

Now consider fixed routing for the same fully connected network with N nodes. Again, suppose all one-link routes and all two-link routes with adjacent links are employed. This again gives $N(N-1)^2/2$ routes. It is easily seen that the computational effort for one iteration of the repeated substitution algorithm is $O(N^3 + CN^2)$ and that the memory requirement is $O(N^2 + C)$. Sequential routing in a fully connected network can be seen to have the same computational and memory requirements. *Thus, the computational and memory requirements for least loaded routing are greater than those for fixed and sequential routing.*

We should mention that if the number of alternative routes is limited to less than the maximum possible, then significantly less computation may be needed. For example, suppose that the number of alternative routes per node pair is equal to M, where $M \ll N-2$. In this case, we have (on average) $|S_j| = 2M$, and a straightforward modification of the first algorithm has $O(CN^2M^2)$ computational effort.

B. Truncated Distributions

The approximation schemes for least loaded routing require an amount of computation that is linearly proportional to C, the capacity of the links. In order to minimize this effect, we set $q_j(m) = 0$ for all $m > M_j$, where M_j, the truncation level, changes from iteration to iteration as discussed below. Once the $M_j, j = 1, \cdots, J$ are determined, then the state-dependent arrival rates

$$\alpha_j(m) = \sum_{R \in \mathcal{R}_j} E_{\boldsymbol{q}}[\lambda_R(\boldsymbol{X}) | X_j = m]$$

are calculated only for $m = 0, \cdots, M_j$. Then, a new set of distributions $q_j(m), m = 0, \cdots, M_j$ are obtained from the state-dependent arrival rates via (5).

TABLE I
TEST NETWORK: LIGHT, MODERATE, AND HEAVY TRAFFIC CONDITIONS ARE OBTAINED BY MULTIPLYING THE ABOVE OFFERED TRAFFIC BY 1, 1.2, AND 1.5, RESPECTIVELY

Link	#(Trunks)	Offered Traffic Rate
1,2	36	27.47
1,3	24	6.97
1,4	324	257.81
1,5	48	20.47
1,6	48	29.11
2,3	96	25.11
2,4	96	101.61
2,5	108	76.78
2,6	96	82.56
3,4	12	11.92
3,5	48	6.86
3,6	24	13.25
4,5	192	79.42
4,6	84	83.00
5,6	336	127.11

To obtain the truncation levels $M_j, j = 1, \cdots, J$, we do the following. Before the first iteration, for each link j we consider an Erlang loss system with capacity C_j and with calls arriving at rate a_j (the exogenous arrival rate to the node pair connected directly by link j). We then find the smallest M_j such that

$$\sum_{m=0}^{M_j} q_j(m) > T \tag{14}$$

where T is the truncation factor and $q_j(\cdot)$ is the idle capacity distribution for the Erlang loss system and the truncation factor could be any number near 1; for example, 0.999. We then determine the state-dependent arrival rates $\alpha_j(m)$ and a new set of distributions $q_j(m)$ for $m = 0, \cdots, M_j, j = 1, \cdots, J$ as discussed above. We then obtain new $M_j, j = 1, \cdots, J$ according to (14) and repeat the whole process.

In very light traffic, the truncation method discussed above does not give a substantial savings in CPU time since $M_j \approx C_j$. However, significant savings can be gained in moderate and heavy traffic.

IV. COMPUTATIONAL RESULTS

A. A Six-Node Test Network

We now compare the approximation schemes for LLR with simulation results for a test network. Consider the six-node fully connected network described in Table I; for each pair of nodes, the number of circuits and the offered traffic are specified. Note that the network is highly asymmetric and that the exogenous offered load to the node pair 2, 4 exceeds the number of circuits in its direct link. We consider three cases: light, moderate, and heavy traffic. In the case of light traffic, trunk reservation is not used. In the cases of moderate and heavy traffic, we use trunk reservation with the same

TABLE II
WEIGHTED AVERAGE PERCENT BLOCKING FOR SIX- NODE TEST NETWORK

	Simulation	Approximation	Approximation with Truncation
Light	0.00%	0.00%	0.00%
Moderate	1.33%	0.73%	0.73%
Heavy	10.43%	10.15%	10.30%

TABLE III
PERCENTAGE OF CALLS BLOCKED IN THE MODERATE TRAFFIC FOR SIX- NODE TEST NETWORK

Node Pair	Simulation	Approximation	Approximation with Truncation
1,2	0.28	0.02	0.02
1,3	0.00	0.00	0.00
1,4	0.45	0.02	0.02
1,5	0.00	0.00	0.00
1,6	0.05	0.00	0.00
2,3	0.00	0.00	0.00
2,4	8.87	5.73	5.80
2,5	0.00	0.00	0.00
2,6	2.86	1.21	1.23
3,4	0.02	0.00	0.00
3,5	0.00	0.00	0.00
3,6	0.00	0.00	0.00
4,5	0.00	0.00	0.00
4,6	0.02	0.00	0.00
5,6	0.00	0.00	0.00

trunk reservation level on each link. Trunk reservation levels $r = 4$ and $r = 5$ are used for moderate and heavy traffic, respectively. The data for this test network has been extracted out of [33]. The simulations are run for 100 holding times for heavy traffic, and for 1000 holding times for light and moderate traffic. Five independent replications are run and averaged in all cases. Convergence of repeated substitutions occurs for all of the approximation algorithms and traffic conditions for this six-node network. For all three traffic conditions, a truncation factor 0.999 is used. All calculations were performed on a Sun 4/280.

In Table II, the weighted average percent blocking obtained by simulation is compared with the approximation schemes. The 95% confidence intervals for the simulations are about 0.01%. In light traffic, the approximations all give 0.00% blocking as does simulation (noticable blocking occurs at this loading for other routing schemes; see [33]). In heavy traffic, the approximations slightly underestimate actual blocking. For moderate traffic, we see that there is a fairly big gap between approximate blocking and exact blocking (although not the 2% gap that occurs with the approximation schemes given in [15], [14], [27]).

We also looked at the blocking percentages for each node pair. In light traffic, simulation gives 0.00% blocking for all node pairs, except for node pair 2, 4 for which it gives 0.01% blocking. Each of the approximations gives 0.00% blocking for all node pairs. Tables III and IV give the percent blocking for each node pair for moderate and heavy traffic, respectively. In heavy traffic, the approximation schemes are in fairly close agreement with simulation. For moderate traffic, the approximations are less accurate.

TABLE IV
PERCENTAGE OF CALLS BLOCKED IN THE HEAVY TRAFFIC FOR SIX- NODE TEST NETWORK

Node Pair	Simulation	Approximation	Approximation with Truncation
1,2	8.2	8.3	8.4
1,3	0.3	0.6	0.6
1,4	14.6	14.3	14.4
1,5	0.8	1.1	1.1
1,6	5.7	7.2	7.3
2,3	0.0	0.0	0.0
2,4	32.2	32.1	32.3
2,5	2.6	1.2	1.2
2,6	19.1	18.5	18.5
3,4	7.6	7.2	7.8
3,5	0.0	0.0	0.0
3,6	0.8	0.7	0.7
4,5	0.7	0.9	1.0
4,6	6.4	5.4	6.0
5,6	0.0	0.0	0.0

TABLE V
CPU TIMES IN SECONDS (ITERATIONS IN PARENTHESES) FOR SIX -NODE TEST NETWORK

	Approximation	Approximation with Truncation
Light	2.4 (3)	1.5 (3)
Moderate	13.0 (19)	4.7 (18)
Heavy	18.0 (27)	3.6 (24)

In Table V, we give the CPU time in seconds for each of the approximation techniques. The number of iterations of repeated substitution is also given in parentheses. The iterations are stopped when the maximum change in point-to-point blocking probability is less than 10^{-8}. Note that only three iterations are required for light traffic, whereas as many as 19 and 27 iterations are required in moderate and heavy traffic, respectively. Also note that truncated distributions can reduce CPU time by a factor of 5 in heavy traffic.

B. A 36-Node Test Network

We also investigate the approximation schemes for an asymmetric fully connected network with 36 nodes and an average link capacity of about 80. We again consider three traffic conditions, which we refer to as light, moderate, and heavy. (We do not give all of the data since there is so much of it.) In all three traffic conditions, trunk reservation level $r = 6$ is used on each link. In the case of light traffic, truncation factor 0.99999 is used. In the cases of moderate and heavy traffic, the truncation factor is 0.9999. For each of the three traffic conditions, the simulations are run for 60 million events; statistics are gathered for the last 50 million events in five batches with 10 million events in a batch. Convergence of repeated substitutions occurs for all of the approximation algorithms and traffic conditions for this 36-node network.

Table VI shows the CPU time utilized by the various algorithms for two full iterations (plus the initial iteration). Note that the Second Algorithm reduces CPU time by about a factor of 13, as predicted by the complexity analysis. Note

TABLE VI
CPU TIMES IN SECONDS FOR TWO -FULL ITERATIONS OF REPEATED SUBSTITUTIONS FOR 36-NODE TEST NETWORK

	1st Algo	2nd Algo	2nd Algo with Trunc
Light	2343	182	100
Moderate	2346	183	89
Heavy	2333	182	81

also that truncation further reduces CPU time by about a factor of 2. We can conclude from Table VI that if an approximation scheme is to be imbedded in a design package that computes blocking probabilities repeatedly, then the First Algorithm is inappropriate.

Table VII presents the CPU times and the weighted average blocking percentages for the Second Algorithm and for the Second Algorithm with truncation. The number of iterations of repeated substitution is also given in parentheses. The iterations are stopped when the maximum change in point-to-point blocking probability is less than 10^{-4}. Note that only 22 iterations are required for heavy traffic, whereas as many as 55 iterations are required in light traffic. (We observed that the truncation factor, either 0.99999 or 0.9999, has little effect on the weighted average blocking percentages.) Note that truncation has reduced the CPU time by a factor of 3 to 4.

Now consider the accuracy of the approximation for the 36-node network. In our various experiments (not all discussed here), we noticed that accuracy improves as the number of nodes increases. However, even for a network with a large number of nodes, there seems to be a narrow "critical region" for the offered loads in which the approximation can be inaccurate. In the 36-node experiments, the "light," "moderate," and "heavy" traffic conditions are chosen in order to highlight the behavior of the approximation in this critical region.

Table VII also gives an overview of the accuracy of the approximation for the 36-node network. In light traffic, the approximation underestimates blocking, although blocking occurs very rarely. In moderate and heavy traffic, the approximation slightly underestimates actual blocking. (Note that the offered loads have been chosen so that the blocking probabilities, even for heavy traffic, are small.)

A better understanding of the accuracy of the algorithm can be obtained by looking at the individual node pairs. Tables VIII–X give the percent blocking for 35 node pairs for light, moderate, and heavy traffic, respectively. Note that, in light traffic, the approximation gives poor results for several node pairs. (For example, for the node pair 1-34 simulation gives about 1% blocking whereas the approximation gives 0.02% blocking.) In moderate traffic, the approximation gives results that are either in or close to the corresponding 95% confidence intervals. In heavy traffic, the approximation is in very close agreement with simulation. Although the results are not reported here, we observed that if the offered loads are increased beyond "heavy traffic" for the 36-node network, then the approximation becomes more and more accurate.

V. STATE- DEPENDENT ROUTING WITH MULTIRATE TRAFFIC

We now develop a novel approximation procedure for state-dependent routing with multirate traffic. This procedure can be used to approximate connection-level blocking for asynchronous transfer mode (ATM) networks or call blocking for multirate circuit-switched networks with flexible slot assignment.

Suppose that a call can hold several circuits simultaneously in a link, which would be the case for video or some other wideband service. More specifically, now suppose that class is assigned to a call when admitted into the network, where a class k call has route $R_k \subseteq \{1, \cdots, J\}$, bandwidth requirement b_k, and offered load ρ_k. When a class k call enters the network, it holds b_k circuits in each link $j \in R_k$ for its duration. Let Γ_j be the set of classes that use link j. Let $X_j, m_j, q_j(\cdot), j = 1, \cdots, J, \boldsymbol{m}$, and $\boldsymbol{q}$ be defined as before. Let $\lambda_k(\boldsymbol{m})$ be the rate at which class k calls are set up when the network is in state $\boldsymbol{m}$. Note that $\lambda_k(\boldsymbol{m})$ is specified by the (state-dependent) routing policy. Clearly, $\lambda_k(\boldsymbol{m})$ must satisfy

$$\lambda_k(\boldsymbol{m}) = 0 \qquad \text{if } m_j < b_k \text{ for some } j \in R_k.$$

In order to illustrate the idea, consider again the third example of Section II. Now suppose there are two "services" that request connections between the three node pairs: a narrowband service that requires one circuit point-to-point, and a wideband service that requests $b > 2$ circuits point-to-point. Suppose that the narrowband calls are routed as before. When a wideband call requests a connection between the pair of nodes directly connected by link 1, the call is set up on link 1 if $m_1 \geq b$ and $m_1 \geq \min(m_2, m_3)$. The wideband call is set up on route $\{2, 3\}$ if $\min\{m_2, m_3\} \geq b$ and $\min\{m_2, m_3\} > m_1$. The routing policy for the wideband calls with direct link 2 and direct link 3 are defined analogously. Thus, the routing policy is a multirate version of LLR without trunk reservation. Note that we have four classes associated with each node pair: A narrowband direct-route class, a narrowband indirect-route class, a wideband direct-route class, and a wideband-indirect route class. Thus, there is a total of twelve classes for this example. It is straightforward to write down the rates for $\lambda_k(\boldsymbol{m})$ for each of the twelve classes.

As for the single-rate case, we assume that $X_1, \cdots, X_J$ are mutually independent. Also assume, with m idle circuits on link j, that the time until the next call of class $k \in \Gamma_j$ is set up on link j is exponentially distributed with parameter

$$\alpha_{jk}(m) = E_{\boldsymbol{q}}[\lambda_k(\boldsymbol{X}) | X_j = m]. \tag{15}$$

It remains to determine $q_j(\cdot)$ from $\alpha_{jk}(\cdot), k \in \Gamma_j$. This involves the analysis of a single-link system with C_j servers and $|\Gamma_j|$ classes, where class k calls have a bandwidth b_k and an arrival rate $\alpha_{jk}(\cdot)$ that depends on the number of idle servers. Such a system does not, in general, have a product form solution, so that the algorithms in [39] are inapplicable. Let $\gamma_k(i) = \alpha_{jk}(C_j - i)$. We suggest that $q_j(m)$ be approximated by $q_j(m) = p(C_j - m), m = 0, \cdots, C_j$.

TABLE VII
CPU TIMES IN SECONDS (NUMBER OF ITERATIONS IN PARENTHESES) AND WEIGHTED AVERAGE PERCENT BLOCKING FOR 36-NODE TEST NETWORK

		2nd Algo	2nd Algo with Trunc	Simulation
CPU time (iterations)	Light	5002 (55)	1607 (55)	
	Mod	3509 (39)	835 (39)	
	Heavy	1974 (22)	415 (22)	
Percent Blocking	Light	1.6×10^{-4}	1.6×10^{-4}	$(0.22 \times 10^{-2}, 0.23 \times 10^{-2})$
	Mod	1.20×10^{-2}	1.20×10^{-2}	$(1.61 \times 10^{-2}, 1.64 \times 10^{-2})$
	Heavy	5.64×10^{-2}	5.65×10^{-2}	$(5.83 \times 10^{-2}, 5.90 \times 10^{-2})$

TABLE VIII
PERCENT BLOCKING FOR SOME NODE PAIRS IN LIGHT TRAFFIC FOR 36-NODE TEST NETWORK

Node Pair	Simulation	Approximation with Trunc	Node Pair	Simulation	Approximation with Trunc
1,2	(0.00,0.00)	0.00	1,3	(0.00,0.00)	0.00
1,4	(0.01,0.02)	0.00	1,5	(0.00,0.01)	0.00
1,6	(0.00,0.01)	0.00	1,7	(0.00,0.00)	0.00
1,8	(0.11,0.19)	0.00	1,9	(0.00,0.00)	0.00
1,10	(0.00,0.00)	0.00	1,11	(0.00,0.08)	0.00
1,12	(0.00,0.00)	0.00	1,13	(0.00,0.02)	0.00
1,14	(0.00,0.00)	0.00	1,15	(0.14,0.23)	0.00
1,16	(0.00,0.00)	0.00	1,17	(0.00,0.00)	0.00
1,18	(0.00,0.00)	0.00	1,19	(0.07,0.20)	0.00
1,20	(0.00,0.00)	0.00	1,21	(0.00,0.01)	0.00
1,22	(0.00,0.00)	0.00	1,23	(0.12,0.21)	0.00
1,24	(0.00,0.01)	0.00	1,25	(0.00,0.01)	0.00
1,26	(0.09,0.37)	0.00	1,27	(0.00,0.00)	0.00
1,28	(0.00,0.00)	0.00	1,29	(0.00,0.00)	0.00
1,30	(0.60,0.82)	0.01	1,31	(0.00,0.00)	0.00
1,32	(0.00,0.01)	0.00	1,33	(0.00,0.00)	0.00
1,34	(0.86,1.11)	0.02	1,35	(0.00,0.00)	0.00
1,36	(0.01,0.05)	0.00			

TABLE IX
PERCENT BLOCKING FOR SOME NODE PAIRS IN MODERATE TRAFFIC FOR 36-NODE TEST NETWORK

Node Pair	Simulation	Approximation with Trunc	Node Pair	Simulation	Approximation with Trunc
1,2	(0.07,0.17)	0.10	1,3	(0.00,0.01)	0.00
1,4	(0.48,0.70)	0.41	1,5	(0.00,0.02)	0.01
1,6	(0.06,0.18)	0.09	1,7	(0.00,0.01)	0.00
1,8	(3.35,3.93)	2.64	1,9	(0.12,0.22)	0.07
1,10	(0.00,0.02)	0.00	1,11	(0.46,0.78)	0.79
1,12	(0.00,0.00)	0.00	1,13	(0.42,0.76)	0.72
1,14	(0.00,0.00)	0.00	1,15	(3.74,4.19)	3.23
1,16	(0.00,0.00)	0.00	1,17	(0.09,0.19)	0.11
1,18	(0.00,0.00)	0.00	1,19	(1.82,2.31)	1.63
1,20	(0.00,0.02)	0.02	1,21	(0.18,0.33)	0.22
1,22	(0.00,0.00)	0.00	1,23	(3.71,4.15)	2.18
1,24	(0.33,0.54)	0.15	1,25	(0.00,0.04)	0.00
1,26	(2.83,3.40)	3.80	1,27	(0.00,0.00)	0.00
1,28	(0.06,0.12)	0.11	1,29	(0.00,0.01)	0.00
1,30	(6.51,7.24)	5.90	1,31	(0.00,0.02)	0.00
1,32	(0.32,0.52)	0.37	1,33	(0.00,0.00)	0.00
1,34	(5.20,5.87)	5.01	1,35	(0.00,0.01)	0.01
1,36	(0.55,0.73)	0.66			

TABLE X
PERCENT BLOCKING FOR SOME NODE PAIRS IN TRAFFIC FOR 36-NODE TEST NETWORK

Node Pair	Simulation	Approximation with Trunc	Node Pair	Simulation	Approximation with Trunc
1,2	(1.18,1.79)	2.05	1,3	(0.01,0.05)	0.07
1,4	(3.12,4.17)	3.86	1,5	(0.08,0.29)	0.28
1,6	(0.77,1.13)	1.29	1,7	(0.00,0.01)	0.02
1,8	(13.46,14.03)	13.88	1,9	(1.54,1.89)	1.72
1,10	(0.16,0.24)	0.09	1,11	(4.46,5.54)	7.71
1,12	(0.00,0.00)	0.00	1,13	(2.14,3.12)	3.03
1,14	(0.04,0.08)	0.07	1,15	(13.41,14.31)	14.12
1,16	(0.00,0.02)	0.02	1,17	(1.45,2.43)	2.54
1,18	(0.00,0.01)	0.01	1,19	(6.77,7.67)	7.57
1,20	(0.14,0.36)	0.42	1,21	(1.40,2.40)	2.16
1,22	(0.00,0.01)	0.01	1,23	(13.83,15.09)	14.44
1,24	(2.77,3.59)	3.04	1,25	(0.42,0.61)	0.22
1,26	(9.37,11.92)	14.04	1,27	(0.00,0.00)	0.00
1,28	(0.59,0.79)	0.85	1,29	(0.01,0.09)	0.05
1,30	(16.67,17.25)	17.39	1,31	(0.03,0.11)	0.08
1,32	(3.55,4.31)	4.85	1,33	(0.00,0.01)	0.00
1,34	(12.19,13.40)	12.65	1,35	(0.04,0.14)	0.14
1,36	(2.55,3.62)	3.72			

where $p(\cdot)$ satisfies the following system of equations:

$$ip(i) = \sum_{k \in \Gamma_j} \frac{b_k}{\mu_k} \gamma_k(i - b_k)\, p(i - b_k), \quad i = 1, \cdots, C_j. \tag{16}$$

$$\sum_{i=0}^{C_j} p(i) = 1. \tag{17}$$

Roberts [37] also proposed this approximation for a *single-link system*, assuming that the state-dependent arrival rates take on at most two values for each class. Assuming that all the classes have the same mean holding time, he found the approximation to be very accurate. More recently, Gersht and Lee [9] studied the same approximation for the single-link system. Their numerical testing confirms the accuracy of the approximation when calls have the same mean holding times; however, they observed that the approximation can be inaccurate when the holding times are different. For the case of different holding times, Gersht and Lee modify the single-link approximation (16) by replacing all of the μ_k's for the link by $\overline{\mu}$, where $\overline{\mu}$ denotes the average departure rate and is determined by a repeated substitutions procedure involving only the isolated link. Their extensive empirical testing shows that the approximation procedure is good for a wide range of model parameters. This modification of (16) should also be used to approximate $q_j(\cdot)$ for networks when classes have different mean holding times.

Note that, for the case of Poisson arrivals for each class, (16) becomes the well-known recursive equation for exactly calculating occupancy probabilities for a single link with multirate traffic [21], [38]. We now further motivate the single-link approximation by showing that it is also exact for a class of state-dependent arrival rates.

Theorem 1: Consider a single-link system with C_j circuits and $|\Gamma_j|$ classes of calls. Let class k calls have mean holding time $1/\mu_k$, bandwidth requirement b_k, and arrival rate $\gamma_k(i)$ when i circuits are busy. Suppose that there is a function $\varphi(\cdot)$ and constants $a_k, k \in \Gamma_j$, such that

$$\gamma_k(i) = a_k \frac{\varphi(i + b_k)}{\varphi(i)} \tag{18}$$

for all $k \in \Gamma_j, i = 0, \cdots, C_j$. Let $p(i), i = 0, \cdots, C_j$ be the probability that i circuits are busy in this system. Then, $p(i), i = 0, \cdots, C_j$ is given by the unique solution to (16) and (17).

Proof: Let n_k denote the number of class k calls in the one-link system and let $\boldsymbol{n} := (n_k, k \in \Gamma_j)$. The state space is given by $\Omega = \{\boldsymbol{n} : \boldsymbol{b} \cdot \boldsymbol{n} \le C_j\}$, where $\boldsymbol{b} \cdot \boldsymbol{n} = \Sigma_{k \in \Gamma_j} b_k n_k$. An argument employing the detailed balance equaitons [22] shows that the equilibrium probability of being in state $\boldsymbol{n} \in \Omega$ is

$$\pi(\boldsymbol{n}) = \frac{\varphi(\boldsymbol{b} \cdot \boldsymbol{n}) \prod_{l \in \Gamma_j} \frac{\rho_l^{n_l}}{n_l!}}{G}$$

where $\rho_l = a_l/\mu_l$ and

$$G = \sum_{\boldsymbol{n} \in \Omega} \varphi(\boldsymbol{b} \cdot \boldsymbol{n}) \prod_{l \in \Gamma_j} \frac{\rho_l^{n_l}}{n_l!}.$$

Observe that

$$p(i) = \sum_{\{\boldsymbol{n} : \boldsymbol{b} \cdot \boldsymbol{n} = i\}} \pi(\boldsymbol{n}) = \frac{1}{G} \sum_{\{\boldsymbol{n} : \boldsymbol{b} \cdot \boldsymbol{n} = i\}} \varphi(i) \prod_{l \in \Gamma_j} \frac{\rho_l^{n_l}}{n_l!}.$$

Thus,

$$
\begin{aligned}
ip(i) &= \frac{1}{G} \sum_{\{\boldsymbol{n}:\boldsymbol{b}\cdot\boldsymbol{n}=i\}} (\boldsymbol{b}\cdot\boldsymbol{n})\,\varphi(i) \prod_{l\in\Gamma_j} \frac{\rho_l^{n_l}}{n_l!} \\
&= \frac{1}{G} \sum_{k\in\Gamma_j} b_k \varphi(i) \sum_{\{\boldsymbol{n}:\boldsymbol{b}\cdot\boldsymbol{n}=i\}} n_k \prod_{l\in\Gamma_j} \frac{\rho_l^{n_l}}{n_l!} \\
&= \frac{1}{G} \sum_{k\in\Gamma_j} b_k \frac{\varphi(i)}{\varphi(i-b_k)} \rho_k \sum_{\{\boldsymbol{n}:\boldsymbol{b}\cdot\boldsymbol{n}=i-b_k\}} \varphi(i-b_k) \prod_{l\in\Gamma_j} \frac{\rho_l^{n_l}}{n_l!} \\
&= \sum_{k\in\Gamma_j} \frac{b_k}{\mu_k} \gamma_k(i-b_k)\, p(i-b_k)
\end{aligned}
$$

which completes the proof.

Thus, when condition (18) holds, then (16)–(17) is exact. When (18) does not hold, then (16)–(17) is an approximation.

We point out that, in the case of fixed routing, the reduced load approximation employing (15) and (16) becomes the knapsack approximation studied in [7], [4].

VI. Concluding Remarks

Is the reduced load approximation an appropriate tool for designing large loss networks with state-dependent routing? Our computational experiments seem to indicate that the procedure gives good ballpark estimates of blocking probabilities; in particular, the estimates appear to be more accurate than those given in [15], [14], [27]. However, we also feel that the procedure should be used with caution since there is a critical region for the loadings in which the accuracy of the approximation may not be acceptable. Thus, discrete-event simulation may be needed to take a ballpark design to final design.

Another important issue concerns the computational requirements of the approximation. Recall that the implementation that holds the most promise has $O(CT^3)$ computational effort and memory requirements. This means that if the number of nodes is doubled, the computational effort and memory requirements are going to grow by a factor of about 8. If the approximation is used with a 108-node network, then the run times are going to take about 27 times longer than those for the 36-node test network. This may be considered excessive as part of an iterative network design procedure. One should also keep in mind that large loss networks can be simulated quite efficiently, with both sequential [36], [35] and parallel [8] implementations.

There are several areas of research that merit further investigation. First, it is of interest to develop parallel implementations of the algorithm for an SIMD computer such as the Connection Machine. Indeed, the approximation scheme can be naturally mapped onto a multiprocessor system where one processor is associated with each link j. In addition to the truncation procedure discussed in Section III, it would be of interest to incorporate the "warm start" idea of [31] in the code. It would then be of interest to compare the parallel implementation (including these coputational features) with discret-event simulation.

It is also of interest, in the context of the reduced load approximation, to investigate the sensitivity of network performance with respect to changes in the offered load and link capacity [24]. In particular, accuracy and computational effort of approximation schemes for sensitivity should be considered.

VII. Acknowledgment

The authors would like to thank J. Ash, J. Chandramohan, D. Mitra, J. Roberts, W. Whitt, and the referees for their comments.

References

[1] J. M. Akinpelu, "The overload performance of engineered networks with nonhierarchical and hierarchical routing," *AT&T Bell Labs Tech. J.*, vol. 63, pp. 1261–1281, 1984.

[2] G. R. Ash, J.-S. Chen, A. E. Frey, and B D. Huang, "Real-time network routing in a dynamic class-of-service network," in *Proc. 13th ITC*, Copenhagen, Denmark, 1991.

[3] V.P. Chaudhary, K.R. Krishnan, and C.D. Pack, "Implementing dynamic routing in the local telephone companies of the USA," in *Proc. 13th ITC*, Copenhagen, Denmark, 1991.

[4] S.P. Chung and K.W. Ross, "Reduced load approximations for multirate loss networks," to appear in *IEEE Trans. Commun.*

[5] R.B. Cooper and S. Katz, "Analysis of alternate routing networks with account taken of nanrandomness of overflow traffic," Tech. Rep., Bell Telephone Lab. Memo., 1964.

[6] E.V. Denardo and H. Park, "Efficient routing of telecommunications traffic," Yale Univ. Tech. Rep., 1991.

[7] Z. Dziong and J. W. Roberts, "Congestion probabilities in a circuit-switched integrated services network," *Perform. Eval.*, vol. 7, pp. 267–284, 1987.

[8] S.G. Eick, A.G. Greenberg, B.D. Lubachevsky, and A. Weiss, "Synchronous relaxation for parallel simulations with applications to circuit-switched networks," to appear in *ACM Trans. Modeling and Simulat.*

[9] A. Gersht and K.J. Lee, "A bandwidth management strategy in ATM networks," GTE Lab. Tech. Rep., 1990.

[10] R. J. Gibbens, P. J. Hunt, and F. P. Kelly, "Bistability in communication networks," in *Disorder in Physical Systems*, G. R. Grimmett and D. J. A. Welsh, Eds. Oxford, England: Oxford Univ. Press, 1990, pp. 113–127.

[11] R.J. Gibbens and F.P. Kelly, "Dynamic routing in fully connected networks," *IMA J. Mathematic Contr. and Inform.*, vol. 7, pp. 77–111, 1990.

[12] R.J. Gibbens, F.P. Kelly, and P.B. Key, "Dynamic alternative routing—Modeling and behaviour," in *Proc. 12th ITC*, Torino, Italy, 1988.

[13] R. J. Gibbens and P. A. Whiting, "An investigation of the accuracy of the implied cost methods of cs network optimization," in *Proc. 5th UK Teletraffic Symp.*, Ashton Univ., 1989.

[14] A. Girard, *Routing and Dimensioning in Circuit-Switched Networks*. Reading, MA: Addison Wesley, 1990.

[15] A. Girard and M.A. Bell, "Blocking evaluation for networks with residual capacity adaptive routing," *IEEE Trans. Commun.*, vol. 37, pp. 1372–1380, 1990.

[16] P. J. Hunt, "Implied costs in loss networks," *Advances in Appl. Prob.*, vol. 21, pp. 661–680, 1989.

[17] P.J. Hunt,, "Limit theorems for stochastic loss networks," Ph.D. thesis, Univ. Cambridge, 1990.

[18] P.J. Hunt and F.P. Kelly, "On critically loaded loss networks," *Advances in Appl. Prob.*, vol. 21, pp. 661–680, 1989.

[19] B.R. Hurley, C.J. Seidl, and W.F. Sewell, "A survey of dynamic routing methods for circuit-switched traffic," *IEEE Commun. Mag.*, vol. 25, pp. 13–21, 1987.

[20] S.S. Katz, "Statistical performance analysis of switched communication networks," in *Proc. 5th ITC*, New York, NY, 1967.

[21] J.S. Kaufman, "Blocking in a shared resource environment," *IEEE Trans. Commun.*, vol. COM-29, no. 10, pp. 1474–1481, 1981.

[22] F.P. Kelly, *Reversibility and Stochastic Networks*. New York: Wiley, 1979.

[23] F.P. Kelly, "Blocking probabilities in large circuit-switched networks," *Adv. in Appl. Prob.*, vol. 18, pp. 473–505, 1986.

[24] F.P. Kelly, "Routing and capacity allocation in networks with trunk reservation," *Math. of Operat. Res.*, vol. 15, pp. 771–792, 1990.

[25] F.P. Kelly, "Loss networks," *The Annals of Appl. Prob.*, vol. 1, pp. 319–378, 1991.

[26] P.B. Key, "Optimal control and trunk reservation in loss networks," *Prob. in Eng. and Inform. Sci.*, vol. 4, pp. 203–242, 1990.

[27] K.R. Krishnan, "Performance evaluation of networks under state-dependent routing," in *Proc. Bellcore Symp. Perform. Model.*, May 1990 and ORSA/TIMS Conf., Philadelphia, PA, Oct. 1990.
[28] K.R. Krishnan and T.J. Ott, "State-dependent routing for telephone traffic: Theory and results," in *Proc. 25th IEEE Contr. and Decision Conf.*, Athens, Greece, 1986, pp. 2124–2128.
[29] D. Mitra, "Asymptotic analysis and computational methods for a class of simple, circuit-switched networks with blocking," *Adv. Appl. Prob.*, vol. 19, pp. 219–239, 1987.
[30] D. Mitra and R.J. Gibbens, "State-dependent routing on symmetric loss networks with trunk reservation, Part II: Asymptotics, optimal design," *Annals of Operat. Res.*, vol. 35, pp. 3–30, 1992.
[31] D. Mitra, R.J. Gibbens, and B.D. Huang, "State-dependent routing on symmetric loss networks with trunk reservation, Part I," to appear in *IEEE Trans. Commun.*
[32] D. Mitra, R. J. Gibbens, and B. D. Huang, "Analysis and optimal design of aggregated-least-busy-alternative routing on symmetric loss networks with trunk reservation," in *Proc. 13th ITC*, Copenhagen, Denmark, 1991.
[33] D. Mitra and J.B. Seery, "Randomized and deterministic routing strategies for circuit-switched networks: Design and performance," *IEEE Trans. Commun.*, vol. 39, pp. 102–116, 1991.
[34] T.J. Ott and K.R. Krishnan, "Seperable routing: A scheme for state dependent routing of circuit switched traffic," *Annals of Operat. Res.*, vol. 35, pp 43–68, 1992.
[35] M. Prindiville, M. Rajasekaren, and K.W. Ross, "Efficient sequential simulation of large-scale loss networks," Tech. Rep., Dept. Comput. and Inform. Sci., Univ. Penn., 1991.
[36] S. Rajasekaran and K.W. Ross, "Fast algorithms for generating discrete random variates with changing distributions," Tech. Rep., Dept. Comput. and Inform. Sci., Univ. Penn., 1992.
[37] J. Roberts, "Teletraffic models for the Telecom 1 integrated services network," in *Proc. 10th ITC*, Montreal, Canada, 1983, paper 1.1-2.
[38] J.W. Roberts, "A service system with heterogeneous user requirements," *Performance of Data Communications Systems and Their Applications*, G. Pujolle, Ed. Amsterdam, The Netherlands: North Holland, 1981, pp. 423–431.
[39] D. Tsang and K.W. Ross, "Algorithms for determining exact blocking probabilities in tree networks," *IEEE Trans. Commun.*, vol. 38, pp. 1266–1271, 1990.
[40] W. Whitt, "Blocking when service is required from several facilities simultaneously," *AT&T Tech. J.*, vol. 64, pp. 1807–1856, 1985.
[41] E. W. M. Wong and T. S. Yum, "Maximum free circuit routing in circuit-switched networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, 1990.
[42] S. Zachary, "On blocking in loss networks," *Adv. Appl. Prob.*, vol. 23, pp. 355–372, 1991.
[43] I.B. Ziedins and F.P. Kelly, "Limit theorems for loss networks with diverse routing," *Adv. Appl. Prob.*, vol. 21, pp. 804–830, 1989.

Shun-Ping Chung, photograph and biography not available at the time of publication.

Arik Khasper, photograph and biography not available at the time of publication.

Keith W. Ross, photograph and biography not available at the time of publication.