TECHNICAL RESEARCH REPORT

Fixed Point Approximation for Multirate Multihop Loss Networks with Adaptive Routing

by Mingyan Liu, John S. Baras

CSHCN T.R. 99-21
(ISR T.R. 99-44)

Fixed Point Approximation for Multirate Multihop Loss Networks with Adaptive Routing

Mingyan Liu, John S. Baras
Center for Satellite and Hybrid Communication Networks
University of Maryland
College Park, MD 20742 *

Abstract

In this paper, we consider a class of loss networks where multiple traffic classes are present, each has different bandwidth requirement, and each traffic stream is routed according to an adaptive routing scheme. The performance metric of interest is the end-to-end call blocking probability. Blocking probabilities in a loss network have been studied quite extensively but very few considered multiple traffic classes and rates together with adaptive/state dependent routing. We propose a fixed-point method, a.k.a. reduced load approximation, to estimate the end-to-end blocking probability in a multihop multirate loss network with adaptive routing. Simulation results are provided to compare with that of approximations. The approximation scheme is shown to be asymptotically correct in a natural limiting regime, and it gives conservative estimates of blocking probabilities under heavy traffic load.

*This work was supported by the Center for Satellite and Hybrid Communication Networks, under NASA cooperative agreement NCC3-528

1 Introduction

This paper is focused on the evaluation of end-to-end call blocking probabilities in a class of loss networks. A loss network is a circuit switched network, where a call requires a fixed amount of bandwidth on every link on a path between the source and destination. A more detailed definition of loss network can be found in [1]. If the network has the required bandwidth on those links when the request arrives, the call is admitted and it will be holding the requested capacities for some time; otherwise the call is rejected. The blocking probability associated with a loss network is the probability that a call finds the network unavailable when it arrives and is thus rejected. A telephone system is a typical loss network. An ATM network can also be viewed as a loss network and the connection level blocking probabilities can be calculated by applying the concept of effective bandwidth [2, 3].

The Erlang formula:

$$E(\nu, C) = \frac{\nu^C}{C!}[\sum_{n=0}^{C} \frac{\nu^n}{n!}]^{-1}, \tag{1}$$

established the loss probability of a single link with C units of bandwidth and calls arrive as a single Poisson process with rate ν. Analytically, when there are multiple links and multiple call classes, with different arrival rates, different bandwidth requirements, and a fixed route associated with each source-destination node pair, a loss network can be modeled as a multidimensional Markov process with the dimension of the process state space being the product of the number of routes allowed in the network and the number of service/call classes. This is because the number of calls of each class on each feasible route uniquely define the state of the network. When alternative routes are present in addition to fixed routes, the Markov process no longer has a product form, and the equilibrium state probabilities can be obtained by writing out the whole set of detailed balance equations and solving them [1]. However, this approach is not practical in dealing with large networks with hundreds of thousands of routes and integrated services with multiple service rates, since the computational complexity is both exponential in the number of routes and exponential in the number of service classes. This leads to the need for the development of computational techniques that provide accurate estimates within reasonable time frame.

The reduced load approximation, also called the Erlang fixed-point method, has been proposed for this scenario and has been studied intensively [1, 4, 5, 6]. The reduced load approximation is based on two assumptions:

(a) *link independence assumption* Blocking occurs independently from link to link.

(b) *Poisson assumption* Traffic flow to each individual link is Poisson and the corresponding traffic rate is the original external offered rate thinned by blocking on other links on the path, thus called the reduced load.

Consider the single rate case. Using Erlang's formula, the blocking probability of each link can be expressed by the offered traffic rate and the blocking probabilities of other links. This leads to a set of nonlinear fixed point equations with the link blocking probabilities as the unknown variables. Solving these equations gives us the approximation on blocking probabilities of each link. The reduced load approximation can be extended to the multi-rate traffic case with sequential alternative routing with trunk reservation, or with dynamic alternative routing.

Most of the earlier works in fixed-point method either studied the multirate traffic situation with fixed routing, such as the Knapsack approximation and Pascal approximation in [5], or focused on state-dependent routing schemes with single traffic rate [4], or multirate service with single link (resource) [7]. In [6], Greenberg and Srikant proposed a fixed-point method to approximate blocking probabilities in a multirate multihop network using sequential routing, but additional computational effort in solving the associated network reliability problem is needed. It is also a common assumption that there exist a direct or two-hop routes between sources and destinations.

We focus our attention on the evolving integrated service networks which have the following characteristics:

(a) The networks are typically much sparser and have a more hierarchical topology. Thus, the assumption of the existence of a direct route between source and destination nodes does not hold in most instances.

(b) Routes can comprise a much larger number of hops (typically around 5 or 6) and there are typically a large number of possible routes between source and destination nodes.

(c) The presence of different traffic classes characterized by widely varying bandwidth requirements and different mean holding times must be considered.

Motivated by the above, we propose to use adaptive routing in combination with the fixed-point method to calculate call blocking probabilities. Most of our work is motivated by [6] and [8]. The organization of the paper is as follows: The next section describes the network model and the adaptive routing schemes proposed for the approximation. Section 3 is the proposed fixed-point approximation method. Asymptotic correctness analysis is given out in Section 4 and In Section 5 we present approximation results compared to simulation results. Section 6 concludes the paper.

2 Network Model

Consider a network with N nodes and J links, each indexed by j. C_j denotes the capacity of link j, in unit bandwidth/circuit. R is the set of all node pairs, each indexed by r. The total number of node pairs is thus $N(N-1)/2$. For each node pair r, there is an associated set of M ordered routes, each indexed by m, representing the m^{th} route in that set. Therefore, pair (r, m) uniquely defines a specific route. The network supports a total of S classes of traffic, indexed by s, and thus (r, s) uniquely defines a specific incoming call request. The bandwidth requirement of such a call on link j is denoted by b_{js}. Note that for different node pairs the classification of calls does not have to be the same. So strictly speaking calls (r_1, s) and (r_2, s) can have different bandwidth requirement on a same link if we allow r_1 and r_2 to have different sets of traffic classes. However, we choose to use this notation because a single uniform classification can always be achieved by increasing the number of classes.

Calls arrive at the network as a Poisson process with an offered load λ_{rs}. A call is accepted if some route has available bandwidth on each of its links to accommodate this call, and the call is routed on that route and holds the bandwidth for a duration with mean time μ_{rs}. If none of the routes are available, the call is rejected. The end-to-end blocking probability of a call (r, s) is denoted by B_{rs}. Throughout this paper the links are considered to be duplex and bi-directional. We use trunks, units of circuits and units of bandwidth interchangeably.

The type of call admission control considered in our model is *trunk reservation*. As Kelly pointed out in [1], if alternative routes use more network resources than first-choice routes, then allowing a blocked call to attempt an alternative route may actually increase the loss probability of a network, and this effect may become even more pronounced if a blocked call can attempt a sequence of alternative routes. An explanation for this phenomenon is that if a link accepts an alternatively routed call, it may later have to block a directly routed call which will then attempt to find a two-hop or multihop route elsewhere in the network. A natural response would be for the link to reject an alternatively routed call if the free circuits on the link are below a certain level. This is the call admission control of the trunk reservation type. An attempting alternatively routed call is only accepted if on each link of the alternative route the number of occupied circuit is less than $C_j - b_{js} - r_s$ where r_s is the *trunk reservation parameter* and may vary with link and class.

The common routing policies which have been studied are fixed routing, alternative routing, sequential alternative routing and adaptive alternative routing. We focus on the last. One important scheme of this kind is called the *Least Loaded Routing* (LLR), where the call is first tried on the direct route, if there is one. If it cannot be setup 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 was implemented in the AT&T long-distance domestic network [1].

A direct extension of LLR to networks where routes tend to have a larger number of hops instead of direct or two-hop routes is a min-max scheme: Pick the link which has the minimum free bandwidth for each route, then pick the route which has the maximum free bandwidth on this link.

Each source-destination node pair r is given a list of alternative routes M_r. When a call arrives, each of the route on the list is evaluated to determine the number of free circuits on its links.

Let C_j^f denote the free/available bandwidth on link j when the call of type (r, s) arrives and consider a route (r, m). Then the route is in a state of admitting this call if and only if

$$C_j^f \geq b_{js}, \text{for all} j \in (r, m)$$

under no trunk reservation admission control, or

$$C_j^f \geq b_{js} + r_s, \text{for all} j \in (r, m)$$

under trunk reservation admission control.

Consider a route (r, m) which is presently available for a type (r, s) call, the *most congested link on the route* is defined as the link with the fewest free circuits on this route:

$$\mathcal{L}_{rm} = argmin_{j \in (r,m)} C_j^f. \tag{2}$$

When there are more than one route available in the alternative route set, the one with the maximum free bandwidth on its most congested link is selected for accepting the call. If none of the routes are admissible, then the call is blocked.

This maximal residual capacity routing scheme tries to avoid bottlenecks on a route. However, while choosing the route which has the most free bandwidth, we might end up taking the longer or the longest routes in the available set and thus using more network resources. This could eventually force calls arriving later to be routed on their longer/longest route as well. Therefore, using trunk reservation along with this routing scheme is a valid choice especially when traffic is heavy.

A more general way of deciding the routing can be expressed as a cost function which takes into account both the length of the route and the congestion level of the route. (For optimization on routing and blocking, Mitra and Morrison present an elaborate form of network revenue in [3] and investigate network optimization problem. Our focus here is limited to admission control.) A simple cost function for route (r, m) could be:

$$w_1 \sum_{j \in (r,m)} b_{js} - w_2 C_{\mathcal{L}_{rm}}^f \tag{3}$$

where the first term is the total number of bandwidth that would be occupied if the route is chosen, and the second term indicates the level of congestion on the route. w_1 and w_2 are weighting parameters. The route which minimizes this cost is chosen. Clearly a longer route will increase the cost. And if w_1 is zero, this becomes the min-max routing we just described.

3 The Fixed-point Method

The fixed point is achieved by mappings between the following four sets of unknown variables:

ν_{js}: the reduced load/arrival rate of class-s calls on link j;

a_{js}: the probability that link j is in a state of admitting class-s calls;

$p_j(n)$: the steady state occupancy probability distribution of link j, i.e., the probability that exactly n units of circuits are being used on link j. n takes any integer value between 0 and C_j, the capacity of link j;

q_{rms}: the probability that a call request (r, s) is attempted on route (r, m). This originates from the fact that different routes have different levels of congestion.

First, we fix a_{js} and q_{rms} to get ν_{js}. Then we let ν_{js} be fixed to get $p_j(n)$ and a_{js}. Finally we fix $p_j(n)$ to get q_{rms}. By repeated substitution, the equilibrium fixed point can be solved for all four sets of unknowns. The mappings are illustrated in the figure bellow:

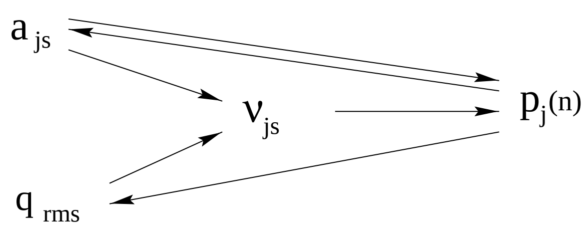

Figure 1: Mappings between variables

A. Mapping 1: a_{js}, $q_{rms} \longrightarrow \nu_{js}$
Define ν_{jrms} as the arrival rate on link j contributed by traffic (r, s) on route (r, m), given that link j is in a state of admitting a class-s call. Then it is given by the reduced load approximation as:

$$\nu_{jrms} = \lambda_{rs} q_{rms} I[j \in (r, m)] \prod_{i \in (r,m), i \neq j} a_{is} \tag{4}$$

where I is the indicator function. The aggregated load of class-s calls on link j is given by

$$\nu_{js} = \sum_{(r,m)} \nu_{jrms}. \tag{5}$$

B. Mapping 2: $\nu_{js} \longrightarrow a_{js}$, $p_j(n)$
Given ν_{js}, we can compute the link occupancy probabilities $p_j(n)$ for each link in the network. This can be done by either using Kaufman's simple recursion [9] when there is no trunk reservation present, or using approaches proposed by Bean, Gibbens and Zachary in [10] and

[7] as suggested by Greenberg in [6]. By the link independence assumption, this mapping is conducted on a per-link basis, and each link is calculated separately and similarly.

In the absence of admission control, classical product-form holds for describing the equilibrium call blocking probabilities [1]. In [9], Kaufman gives a simple one dimensional recursion for calculating the link occupancy distribution probabilities. Let n_s denote the number of class-s calls in progress on this link and b_{js} is the bandwidth requirement of class-s calls on link j. Then for link j we have

$$np_j(n) = \sum_s b_{js} \frac{\nu_{js}}{\mu_{rs}} p_j(n - b_{js}), \quad n = 1, ..., C_j, \tag{6}$$

where $p_j(n) = 0$ if $n < 0$ and

$$\sum_{n=0}^{C_j} p_j(n) = 1. \tag{7}$$

Also it's easy to see that $n = \sum_s b_{js} n_s$.

The probability that a class-s call is admitted to link j is given by

$$a_{js} = 1 - \sum_{n=C_j - b_{js}+1}^{C_j} p_j(n) = \sum_{n=0}^{C_j - b_{js}} p_j(n). \tag{8}$$

Admission control destroys the product form of the link occupancy probabilities $p_j(n)$, which in turn destroys the efficient exact computation of those probabilities just described. To solve for these probabilities, we need to solve for the equilibrium distribution of the associated Markov chain, whose state space is a lattice embedded in the simplex $\sum_s b_{js} n_s \leq C$, $n_s \geq 0$. The computational cost is prohibitive, even for moderate C and $S = 2$. A method to compute the aggregated occupancy probabilities $p(n)$ at a cost linear in C is needed. Approaches proposed in [10, 7] transform the problem into a one-dimensional one by assuming that while n_s, the number of calls in progress of a class s varies, the proportion of such calls in progress remains fixed (or varies slowly).

In [6], the following method is used. Let α_{js} denote the average number of calls of type s in progress on link j,

$$\alpha_{js} = a_{js} \nu_{js} / \mu_{rs}, \tag{9}$$

since calls enter into service at rate $a_{js} \nu_{js}$ and depart at rate $\alpha_{js} \mu_{rs}$.

Consider the one-dimensional Markov chain, for any given state n and call class s, with the following state transition rates:
From state n to state $n + b_{js}$, $\nu_{js} I(C_j - n \geq r_s + b_{js})$;

From state n to state $n - b_{js}$, $\mu_{rs} n \frac{\alpha_{js}}{\sum_t \alpha_{jt}} I(n \geq b_{js})$.
The probability of admitting a call of class s is given by

$$a_{js} = 1 - \sum_{n=C_j - b_{js} - r_s + 1}^{C_j} p_j(n) = \sum_{n=0}^{C_j - b_{js} - r_s} p_j(n). \tag{10}$$

Note that $p_j(n) \longrightarrow \alpha_{js} \longrightarrow p_j(n)$ forms another fixed-point problem, which can be solved by iteration to get the equilibrium distribution $p_j(n)$ and thus a_{js}.

If we use the cost function presented in the previous section to make routing decisions, trunk reservation scheme will no longer be necessary since the idea of trunk reservation admission control is to prevent routing calls onto those longer routes and the cost function has already taken the length of the route into consideration.

C. Mapping 3: $p_j(n) \longrightarrow q_{rms}$
Given $p_j(n)$, define for link j, the probability of no more than n trunks are free (at most n trunks are free) as:

$$t_j(n) = \sum_{k=0}^{n} p_j(C_j - k). \tag{11}$$

Consider the case of no trunk reservation admission control, and use the min-max routing scheme extended from LLR described in the previous section, the probability of attempting a call of (r, s) on route (r, m) is the probability that all routes before the m^{th} route on the routing list have fewer free trunks on their most congested links, and all routes after the m^{th} route have at most the same number of free trunks on their most congested links, which can be expressed as:

$$q_{rms} = \sum_{n=1}^{C_{\mathcal{L}_{rm}}} p_{\mathcal{L}_{rm}}(C_{\mathcal{L}_{rm}} - n) \prod_{k=1}^{m-1} t_{\mathcal{L}_{rk}}(n-1) \prod_{k=m+1}^{M} t_{\mathcal{L}_{rk}}(n). \tag{12}$$

We consider steady state and the free bandwidth on link j C_j^f is replaced by $E[C_j^f]$, the expected average free capacity. Thus $\mathcal{L}_{rm}$, the most congested link on a route, becomes the statistically most congested link as:

$$\mathcal{L}_{rm} = argmax_{j \in (r,m)} z_j \tag{13}$$

where z_j is defined as the link load of link j:

$$z_j = \sum_{rms} \frac{\nu_{jrms} b_{js}}{\nu_{rs} C_j}, \tag{14}$$

which is also the long term average of link utilization.

When there is admission control with trunk reservation parameter r_s, this probability becomes:

$$q_{rms} = \sum_{n=1}^{C_{\mathcal{L}rm}} p_{\mathcal{L}_{rm}}(C_{\mathcal{L}_{rm}} - n) t_{\mathcal{L}_{r1}}(n-1) \prod_{k=2}^{m-1} t_{\mathcal{L}_{rk}}(n + r_s - 1) \prod_{k=m+1}^{M} t_{\mathcal{L}_{rk}}(n + r_s) \qquad (15)$$

assuming that we do not impose trunk reservation on the first route in a set, since naturally that would be the shorted one among all even if it is not the direct route.

If we use the cost function proposed in the previous section, then since the choice of m^{th} route for routing the call indicates that all the routes before m^{th} route have a higher cost than the m^{th} route, and all the routes after the m^{th} route have at least the same cost, the probability of attempting the call on the m^{th} route can be expressed as:

$$\begin{aligned} q_{rms} \quad &= \quad \sum_{n=1}^{C_{\mathcal{L}rm}} p_{\mathcal{L}_{rm}}(C_{\mathcal{L}_{rm}} - n) \prod_{k=1}^{m-1} t_{\mathcal{L}_{rk}}(\frac{w_1}{w_2} \sum_{j \in (r,k)} b_{js} - \frac{w_1}{w_2} \sum_{j \in (r,m)} b_{js} + n - 1) \cdot \\ &\prod_{k=m+1}^{M} t_{\mathcal{L}_{rk}}(\frac{w_1}{w_2} \sum_{j \in (r,k)} b_{js} - \frac{w_1}{w_2} \sum_{j \in (r,m)} b_{js} + n). \end{aligned} \qquad (16)$$

D. End-to-end blocking probabilities

Finally, the end-to-end blocking probability for calls of class s between source-destination node pair r is given by

$$B_{rs} = 1 - \sum_m q_{rms} \prod_{j \in (r,m)} a_{js}. \qquad (17)$$

Repeated substitution is used to obtain the equilibrium fixed point. And the end-to-end blocking probabilities can be calculated from the fixed point.

4 Asymptotic Correctness

By using Brouwer's fixed point theorem, it's easy to show that there exists a fixed point under the proposed fixed point approximation. In this section, we analyze the asymptotic correctness of our fixed point approximation. First we give a steady state explanation for q_{rms}, and formulate an optimization problem to solve for the most probable state for a single link. Then we establish an limiting regime and show that under the specified limiting regime, the blocking probability converges to our fixed point approximation.

We make the following observations. Under steady state, for traffic stream (r, s), the probability of attempting the call on the m^{th} route is q_{rms}. In reality, when a call request comes, using an adaptive routing scheme, the call is routed according to the actual traffic load in the network at that point in time. This type of traffic dispersement is called "metering" [11]. However, in our approximation the routing is modeled as if that each traffic stream has fixed probabilities to be routed onto a set of routes, and those probabilities add up to 1. This method is called "randomization". The metering method generally gives a better performance over randomization. Therefore, our approximation represents a conservative estimate, especially under heavy traffic, of the end-to-end call blocking probabilities.

Accepting this assumption, since random splitting Poisson process according to a fixed probability distribution results in processes which are individually Poisson, we have $\nu_{rms} = \lambda_{rs} \cdot q_{rms}$, the equivalent offered load onto route (r, m) from traffic (r, s), which is still a Poisson process.

Define vector $\mathbf{n} = \{n_{rms}\}$, where n_{rms} is the number of calls in progress on route (r, m) from traffic stream (r, s). For clearer notation purposes, let b_{jrms} be the bandwidth requirement on link j from call (r, s) on route (r, m), and define vector $\mathbf{b} = \{b_{jrms}\}$. Also define vector $\mathbf{C} = \{C_j\}$ to be the link capacity. For a single link the stationary distribution $\pi(\mathbf{n})$ is given by:

$$\pi(\mathbf{n}) = G(\mathbf{n})^{-1} \prod_{(r,m)} \prod_{s} \frac{\nu_{rms}^{n_{rms}}}{n_{rms}!}, \quad \mathbf{n} \in A(\mathbf{C}), \tag{18}$$

where

$$A(\mathbf{C}) = \{\mathbf{n} > \mathbf{0} : \mathbf{b} \cdot \mathbf{n} \leq \mathbf{C}\} \tag{19}$$

defines the set for all feasible $\mathbf{n}$ under the link capacity constraint, and $G(\mathbf{n})$ is the normalizing factor:

$$G(\mathbf{n}) = \sum_{\mathbf{n}} (\prod_{(r,m)} \prod_{s} \frac{\nu_{rms}^{n_{rms}}}{n_{rms}!}). \tag{20}$$

Following Kelly's method in [1], we form the optimization problem of maximizing $\pi(\mathbf{n})$ to find the most probable state $\mathbf{n}$:

$$\begin{aligned} &\text{Max} \sum_{(r,m)} \sum_{s} (n_{rms} log \nu_{rms} - log \nu_{rms}!) \qquad (21) \\ &\text{S.t. } \mathbf{n} \geq 0, \ \mathbf{b} \cdot \mathbf{n} \leq \mathbf{C}. \end{aligned}$$

Using Sterling's formula $\log n! \approx n \log n - n$, and replacing $\mathbf{n}$ by real vector $\mathbf{x} = \{x_{rms}\}$, the primal problem becomes:

$$\begin{aligned} &\text{Max} \sum_{(r,m)} \sum_{s} (x_{rms} \log \nu_{rms} - x_{rms} \log x_{rms} + x_{rms}) \qquad (22) \\ &\text{S.t. } \mathbf{x} \geq 0, \ \mathbf{b} \cdot \mathbf{x} \leq \mathbf{C}. \end{aligned}$$

The objective function is differentiable and strictly concave over $x_{rms} \geq 0$; the feasible region is a closed convex set. Therefore there exists a unique maximum. Using Lagrangian method, the maximum can be found to be:

$$x_{rms}(\mathbf{y}) = \nu_{rms} \cdot \exp(-\sum_j y_j b_{jrms}). \tag{23}$$

where $\mathbf{y} = \{y_j\}$ is the Lagrangian multiplier. The constraints become

$$\mathbf{x}(\mathbf{y}) \geq 0, \ \mathbf{C} - \mathbf{b} \cdot \mathbf{x}(\mathbf{y}) \geq 0. \tag{24}$$

By introducing transformed variable

$$d_j = 1 - exp(-y_j) \tag{25}$$

we can rewrite the maximum in (23) as

$$x_{rms} = \nu_{rms} \cdot \prod_{j \in (r,m)} (1 - d_j)^{b_{jrms}}, \tag{26}$$

and d_j is any solution to the following:

$$\sum_{(r,m)} \sum_s b_{jrms} \nu_{rms} \cdot \prod_i (1 - d_i)^{b_{irms}} \begin{cases} = C_j & \text{if} \quad d_j > 0 \\ \leq C_j & \text{if} \quad d_j = 0 \end{cases} \tag{27}$$

and $d_j \in [0, 1)$.

Using the limiting scheme due to Kelly [1], we consider a sequence of networks indexed by N with increasing link capacity and offered traffic load. In addition, we also allow the number of alternative routes for each source-destination node pair to increase with N. This results in the following limiting regime:

$$\begin{aligned} &\frac{\lambda_{rs}(N)}{N} \longrightarrow \lambda_{rs}, \quad \frac{C_j(N)}{N} \longrightarrow C_j \quad \text{as} \ \ N \longrightarrow \infty \ \ \text{and} \\ &\sum_M q_{rms} = 1 \quad M \longrightarrow \infty \end{aligned} \tag{28}$$

with λ_{rs}/C_j fixed, and M is the total number of alternative routes. Let

$$k_{jrms} = \frac{\nu_{rms}}{C_j} = \frac{\lambda_{rs} q_{rms}}{C_j} \tag{29}$$

also be fixed based on our assumption with q_{rms}.

Following from [1], the blocking probability $B_{rms}(N)$, which is the stationary probability that a call from source (r, s) is accepted by route (r, m) is given by:

$$1 - B_{rms}(N) = q_{rms} \prod_j (1 - d_j)^{b_{jrms}} + o(1) \tag{30}$$

where d_j is the solution to (26).

Now consider the link occupancy probability distribution of link j given by the Kaufman recursion (this is a restate of (8)):

$$np_j(n) = \sum_{rms} b_{jrms} \frac{\nu_{jrms}}{\mu_{rs}} p_j(n - b_{jrms}), \quad n = 0, 1, ..., C_j(N). \tag{31}$$

We observe that $d_j = p_j(C_j)$ forms a set of valid solution to (26), which is the probability that the link is fully occupied. Denote n' as the number of free circuits on link j (n is the number of circuits occupied), and p' as the distribution of n'. As $N \longrightarrow \infty$ and $C_j(N) \longrightarrow \infty$, the distribution $p'_j(n') = p_j(C_j(N) - n)$ converges weakly to the geometric distribution given by [12]:

$$p'_j(n') = (1 - p)p^{n'} \tag{32}$$

where p is the positive root of

$$1 = \sum_{rms} \frac{b_{jrms}}{\mu_{rs}} k_{jrms} p^{b_{jrms}}. \tag{33}$$

Substitute $n' = 0$ into (32),

$$p = 1 - p'_j(0) = 1 - p_j(C_j(N)) = 1 - d_j. \tag{34}$$

So the probability that a call from source (r, s) attempted on route (r, m) can be admitted on link j is

$$\begin{aligned} a_{js} &= \sum_{n'=b_{jrms}}^{C_j(N)} p'_j(n') \\ &= (1-p)(p^{b_{jrms}} + p^{b_{jrms}+1} + ... + p^{C_j(N)}) \\ &= p^{b_{jrms}} - p^{C_j(N)+1} \\ &\stackrel{C_j(N)\to\infty}{=} p^{b_{jrms}} \\ &= (1 - d_j)^{b_{jrms}}. \end{aligned} \tag{35}$$

Hence the asymptotic form (30) can be written as

$$\begin{aligned} 1 - B_{rms}(N) &= q_{rms} \prod_j (1 - d_j)^{b_{jrms}} + o(1) \\ &= q_{rms} \prod_j a_{js} + o(1) \end{aligned} \tag{36}$$

Therefore we get the approximation

$$\begin{aligned} B_{rs}(N) &= 1 - \sum_m (1 - B_{rms}(N)) \\ &\stackrel{M,N\longrightarrow\infty}{=} 1 - \sum_m q_{rms} \prod_j a_{js}, \end{aligned} \tag{37}$$

which is the form we presented (17) in the algorithm.

Similarly, from (26), load on route (r,m) from source (r,s) becomes

$$x_{rms} = \nu_{rms} \cdot \prod_{j\in(r,m)} a_{js} = q_{rms}\lambda_{rs} \prod_{j\in(r,m)} a_{js}. \tag{38}$$

Therefore, as seen by any individual link i,

$$x_{irms} = q_{rms}\lambda_{rs} \prod_{j\in(r,m),j\neq i} a_{js}, \tag{39}$$

which is our first mapping in the algorithm.

5 Experiment and Evaluation

In this section we give two network examples to compare the approximation results with that of simulation.

The first example is a five-node fully connected network depicted in Figure 2.

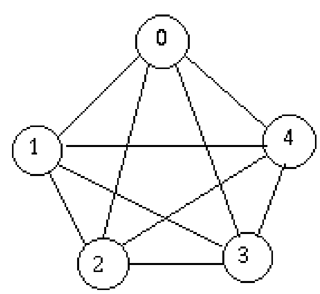

Figure 2: Topology of Example One.

Capacity for each link is set to be 100. There are three classes of connections, which are indexed 1, 2, and 3. They have bandwidth requirements of 1, 2, and 3, respectively. When call admission control is used under heavy traffic, the trunk reservation parameter for each class is 2, 4, and 6, respectively.

For each node pair, the direct route and all two-hop routes are allowed. The direct route is listed first in the routing list, and the two-hop routes are listed in random order.

The medium traffic rates are listed in Appendix A. Heavy traffic rates are set to be double the medium rates. Simulation models were built using OPNET (OPtimized Network

Engineering Tool). Both simulation and fixed point algorithm were run on a SunSparc 20 workstation.

We only display three node pairs here for comparison between fixed-point approximation (FPA) and discrete event simulation (DES). All simulations were run to get a 95% confidence interval. The results are listed in Table 1 through Table 3, with Table 1 showing results for medium traffic, and Tables 2 and 3 for heavy traffic with and without trunk reservation, respectively.

Although the traffic in this case is highly asymmetric, since routing is "symmetric" in the sense that all node pairs are using one direct route and three two-hop routes, each and every traffic stream is imposing on the network in the same way. Therefore, we observe that connections of the same class, with the same bandwidth requirement, encounter approximately the same blocking probability regardless of their source-destination node pair and input rate. However, they do vary slightly from one to another reflecting the random order in which the two-hop routes are listed.

Node Pair	Class	FPA	DES
$(0, 3)$	1	0.223432	(0.0341, 0.0348)
	2	0.398688	(0.2274. 0.2276)
	3	0.535481	(0.4730, 0.4735)
$(1, 2)$	1	0.223938	(0.0358, 0.0360)
	2	0.399227	(0.2209, 0.2212)
	3	0.535798	(0.4701, 0.4710)
$(2, 4)$	1	0.222406	(0.0336, 0.0340)
	2	0.396997	(0.2300, 0.2303)
	3	0.533397	(0.4673, 0.4677)
Number of Iterations		11	
CPU Time(seconds)		1.55	7.1×10^3

Table 1: Ex.1 with Medium Traffic

The second example is borrowed from [6] with minor changes. The topology is derived from an existing commercial network and is depicted in Figure 2 below.

There are 16 nodes and 31 links, with link capacity ranging from 60 to 180 trunks. The detailed link-by-link traffic statistics and link capacities can be found in [6]. For the purpose of self-sufficiency, we are also providing it in Appendix A. The traffic in the network consists of four types, namely class-1, 2, 3, and 4, and require bandwidth of 1, 2, 3, and 4 trunks, respectively. No admission control is employed in this experiment.

In routing, any node pair is allowed routes that have at most 4 hops. Multiple routes for one node pair are listed in order of increasing hops, with ties broken randomly. Each link is considered to have same unit length, so only the hop number is counted.

Node Pair	Class	FPA	DES
$(0, 3)$	1	0.432178	(0.1921, 0.1923)
	2	0.672037	(0.6059, 0.6070)
	3	0.806792	(0.8340, 0.8342)
$(1, 2)$	1	0.429546	(0.1941, 0.1942)
	2	0.668035	(0.6076, 0.6078)
	3	0.802406	(0.8257, 0.8261)
$(2, 4)$	1	0.425318	(0.1849, 0.1851)
	2	0.663327	(0.6002, 0.6004)
	3	0.798391	(0.8290, 0.8291)
Number of Iterations		9	
CPU Time(seconds)		1.24	1.3×10^4

Table 2: Ex.1 with Heavy Traffic and no Trunk Reservation

Node Pair	Class	FPA	DES
$(0, 3)$	1	0.066721	(0.0023, 0.0024)
	2	0.549455	(0.6117, 0.6120)
	3	0.863617	(0.9763, 0.9764)
$(1, 2)$	1	0.065072	(0.0019, 0.0020)
	2	0.544652	(0.6041, 0.6043)
	3	0.855777	(0.9741, 0.9743)
$(2, 4)$	1	0.059249	(0.0014, 0.0016)
	2	0.524917	(0.5765, 0.5766)
	3	0.845543	(0.9709, 0.9711)
Number of Iterations		12	
CPU Time(seconds)		10.86	1.3×10^4

Table 3: Ex.1 with Heavy Traffic and Trunk Reservation

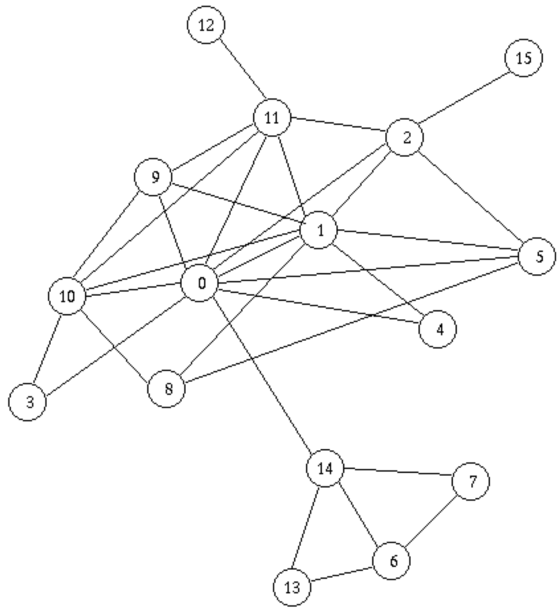

Figure 3: Topology of Example Network

Results for some selected node pairs and classes are listed in Table 4 through 8, each corresponding to a different traffic load. Table 4 corresponds to the "nominal" traffic which is provided in Appendix A. Tables 5 through 8 show the results for traffic 1.4, 1.6 and 1.8 times the nominal traffic, respectively.

The proposed fixed-point approximation gives conservative estimations generally, and it improves as the load gets heavier. These results strengthen the argument that these approximations are indeed very useful as estimators of worst case performance.

Node Pair	Class	FPA	DES
$(0,4)$	4	0.000178	(0.0, 0.0)
$(0,13)$	1	0.006341	(0.0021, 0.0034)
$(1,6)$	1	0.006473	(0.0030, 0.0034)
$(5,6)$	3	0.020463	(0.0189, 0.0201)
$(6,10)$	2	0.013222	(0.0109, 0.0138)
$(9,13)$	4	0.028468	(0.0185, 0.0245)
Number of Iterations		18	
CPU Time(seconds)		94.1	3.7×10^4

Table 4: Ex.2 Nominal Traffic.

Node Pair		Class	FPA	DES
$(0, 4)$		4	0.003234	(0.0, 0.0)
$(0, 13)$		1	0.036512	(0.0351, 0.0369)
$(1, 6)$		1	0.036999	(0.0303, 0.0311)
$(5, 6)$		3	0.114667	(0.1103, 0.1137)
$(6, 10)$		2	0.073531	(0.0543, 0.0573)
$(9, 13)$		4	0.164185	(0.1213, 0.1268)
Number of Iterations			23	
CPU Time(seconds)			120.35	3.9×10^4

Table 5: Ex. 2 1.2 Times The Nominal Traffic.

Node Pair		Class	FPA	DES
$(0, 4)$		4	0.018213	(0.0122, 0.0179)
$(0, 13)$		1	0.074434	(0.0729, 0.0766)
$(1, 6)$		1	0.077371	(0.0697, 0.0701)
$(5, 6)$		3	0.229528	(0.2262, 0.2278)
$(6, 10)$		2	0.147436	(0.1420, 0.1483)
$(9, 13)$		4	0.307191	(0.2794, 0.2848)
Number of Iterations			28	
CPU Time(seconds)			145.43	4.3×10^4

Table 6: Ex. 2 1.4 Times The Nominal Traffic.

Node Pair		Class	FPA	DES
$(0, 4)$		4	0.055354	(0.0512, 0.0549)
$(0, 13)$		1	0.107588	(0.0987, 0.1012)
$(1, 6)$		1	0.117211	(0.1113, 0.1121)
$(5, 6)$		3	0.332202	(0.3137, 0.3142)
$(6, 10)$		2	0.212533	(0.2164, 0.2210)
$(9, 13)$		4	0.424501	(0.3380, 0.3465)
Number of Iterations			24	
CPU Time(seconds)			125.55	5.6×10^4

Table 7: Ex. 2 1.6 Times The Nominal Traffic.

Node Pair	Class	FPA	DES
$(0,4)$	4	0.112658	(0.0025, 0.0026)
$(0,13)$	1	0.135564	(0.1492, 0.1500)
$(1,6)$	1	0.156322	(0.1445, 0.1466)
$(5,6)$	3	0.419781	(0.3922, 0.3940)
$(6,10)$	2	0.269145	(0.2572, 0.2583)
$(9,13)$	4	0.519083	(0.4791, 0.4793)
Number of Iterations		24	
CPU Time(seconds)		125.11	2.3×10^6

Table 8: Ex. 2 1.8 Times The Nominal Traffic.

6 Conclusion

In this paper we presented an approximation scheme of calculating the end-to-end, class-by-class blocking probability of a loss network with multirate traffic and adaptive routing scheme. It provides fairly good estimates of call blocking probabilities under normal and heavy traffic orders of magnitude faster than discrete event estimation. We also presented asymptotic analysis of the fixed point algorithm and showed that this algorithm gives conservative estimates in general.

APPENDIX A
TRAFFIC PARAMETERS

Link j	s	λ_{js}	Link j	s	λ_{js}	Link j	s	λ_{js}	Link j	s	λ_{js}
$(0,1)$	1	20.0	$(1,3)$	1	30.0	$(0,4)$	1	6.0	$(3,4)$	1	51.0
	2	15.0		2	7.0		2	0.0		2	26.0
	3	10.0		3	17.0		3	32.0		3	0.0
$(0,2)$	1	5.0	$(1,4)$	1	3.0	$(2,4)$	1	15.0	$(1,2)$	1	37.0
	2	38.0		2	20.0		2	15.0		2	20.0
	3	9.0		3	20.0		3	20.0		3	5.0
$(0,3)$	1	16.0	$(2,3)$	1	0.0						
	2	17.0		2	15.0						
	3	16.0		3	20.0						

Table 9: Traffic parameters for Example One.

Link j	C_j	Link j	C_j	Link j	C_j	Link j	C_j	Link j	C_j
(0,1)	180	(0,2)	120	(0,3)	120	(0,4)	120	(0,5)	180
(0,9)	120	(0,10)	180	(0,11)	180	(0,14)	180	(1,2)	120
(1,4)	120	(1,5)	120	(1,8)	120	(1,9)	120	(1,10)	120
(1,11)	120	(2,5)	120	(2,11)	120	(2,15)	120	(3,10)	120
(5,8)	60	(6,7)	120	(6,13)	120	(6,14)	120	(7,14)	120
(8,10)	120	(9,10)	120	(9,11)	60	(10,11)	120	(11,12)	60
(13,14)	120								

Table 10: Link Capacities for Example Two.

Link j	λ_{j1}	Link j	λ_{j1}	Link j	λ_{j1}	Link j	λ_{j1}	Link j	λ_{j1}
(1,0)	78.65	(2,0)	0.37	(2,1)	11.19	(3,0)	0.96	(5,0)	0.28
(5,1)	6.33	(5,3)	0.001	(5,4)	0.13	(6,0)	0.37	(6,1)	9.82
(6,5)	0.37	(7,1)	0.001	(8,0)	0.01	(8,1)	0.37	(8,2)	.0001
(8,4)	.0008	(9,0)	0.38	(9,1)	1.12	(9,2)	0.37	(9,5)	0.37
(9,6)	0.78	(9,8)	0.37	(10,0)	0.06	(10,1)	0.37	(11,1)	1.12
(11,2)	0.007	(11,6)	0.01	(11,8)	0.38	(13,0)	1.31	(13,1)	0.38
(13,2)	0.37	(13,6)	.0003	(13,8)	0.37	(13,9)	0.75	(13,10)	0.008
(13,11)	.0001	(14,0)	0.001	(14,1)	0.75	(14,5)	0.07	(14,6)	0.75
(14,13)	.0003	(15,0)	0.37	(15,1)	15.61	(15,2)	0.37	(15,3)	0.37
(15,4)	0.37	(15,6)	0.37	(15,9)	0.37	(15,13)	0.37		

Table 11: Arrival rates for class 1 calls for Experiment 2.

Link j	λ_{j2}	Link j	λ_{j2}	Link j	λ_{j2}	Link j	λ_{j2}	Link j	λ_{j2}
(1,0)	5.97	(2,0)	6.35	(2,1)	3.36	(3,0)	5.97	(3,1)	0.37
(4,0)	1.49	(4,1)	0.37	(4,2)	0.75	(5,0)	28.90	(5,1)	2.24
(5,2)	22.40	(5,3)	2.24	(5,4)	8.96	(6,0)	1.49	(6,1)	1.49
(6,2)	0.75	(6,3)	0.008	(6,5)	3.73	(8,1)	2.24	(8,2)	1.49
(8,5)	5.97	(8,6)	1.50	(9,0)	4.11	(9,1)	9.71	(9,2)	1.87
(9,3)	0.37	(9,4)	0.37	(9,5)	3.37	(9,6)	3.36	(9,8)	4.48
(10,0)	5.99	(10,1)	2.24	(10,2)	0.75	(10,3)	0.07	(10,5)	0.09
(10,6)	0.75	(10,9)	8.96	(11,0)	2.24	(11,1)	5.97	(11,2)	1.49
(11,4)	0.75	(11,5)	2.24	(11,6)	0.75	(11,9)	6.35	(13,0)	1.49
(13,1)	1.87	(13,2)	0.37	(13,4)	0.75	(13,6)	0.08	(13,10)	0.75
(14,0)	1.12	(14,1)	1.12	(14,2)	1.49	(14,5)	8.21	(14,6)	0.75
(14,9)	2.24	(14,10)	1.12	(14,13)	1.12	(15,0)	1.49	(15,1)	0.75
(15,2)	3.14	(15,3)	0.75	(15,4)	0.75	(15,6)	0.75	(15,13)	0.75

Table 12: Arrival rates for class 2 calls for Experiment 2.

Link j	λ_{j3}	Link j	λ_{j3}	Link j	λ_{j3}	Link j	λ_{j3}
(5,1)	0.37	(6,1)	0.37	(6,5)	.0005	(9,0)	0.37
(10,1)	0.37	(13,6)	0.37	(14,1)	.0003		

Table 13: Arrival rates for class 3 calls for Experiment 2.

Link j	λ_{j4}	Link j	λ_{j4}	Link j	λ_{j4}	Link j	λ_{j4}	Link j	λ_{j4}
(1,0)	1.28	(2,0)	0.22	(2,1)	1.60	(3,0)	0.38	(3,1)	1.12
(3,2)	0.75	(4,0)	2.57	(4,1)	2.60	(4,2)	0.37	(5,0)	1.75
(5,1)	1.18	(5,2)	0.13	(5,4)	0.07	(6,0)	1.50	(6,1)	0.52
(6,2)	0.46	(6,3)	0.01	(6,5)	0.01	(7,0)	0.002	(7,2)	0.002
(7,3)	0.001	(7,4)	0.002	(7,6)	.0001	(8,0)	0.001	(8,1)	0.26
(8,5)	0.008	(9,1)	0.37	(9,3)	0.37	(9,5)	0.003	(10,0)	4.22
(10,1)	2.20	(10,2)	0.98	(10,3)	1.07	(10,4)	0.40	(10,5)	1.22
(10,6)	1.62	(10,8)	1.65	(10,9)	0.37	(11,0)	0.76	(11,1)	2.57
(11,2)	1.19	(11,3)	1.12	(11,4)	1.50	(11,6)	0.01	(11,8)	0.37
(11,10)	3.30	(13,0)	1.23	(13,1)	2.24	(13,2)	0.43	(13,3)	0.37
(13,5)	0.50	(13,6)	1.03	(13,7)	0.38	(13,9)	0.37	(13,10)	1.26
(14,0)	1.35	(14,1)	0.07	(14,2)	0.38	(14,5)	0.46	(14,6)	0.41
(14,9)	0.37	(14,10)	0.70	(14,11)	0.75	(14,13)	0.29	(15,1)	0.75
(15,8)	0.37	(15,9)	.0004	(15,10)	0.38	(15,11)	.0001		

Table 14: Arrival rates for class 4 calls for Experiment 2.

References

[1] F. P. Kelly. Loss Networks. *The Annals of Applied Probability*, 1(3):319–378, 1991.

[2] J. Y. Hui. Resource allocation for broadband networks. *IEEE J. Sel. Areas Commun.*, 6:1598–1608, 1988.

[3] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan. ATM Network Design and Optimization: A Multirate Loss Network Framework. *IEEE Trans. Networking*, 4(4):531–543, 1996.

[4] S. Chung, A. Kashper, and K. W. Ross. Computing Approximate Blocking Probabilities for Large Loss Networks with State-Dependent Routing. *IEEE Trans. Networking*, 1(1):105–115, 1993.

[5] S. Chung and K. W. Ross. Reduced Load Approximations for Multirate Loss Networks. *IEEE Trans. Commun.*, 41(8):1222–1231, 1993.

[6] A. G. Greenberg and R. Srikant. Computational Techniques for Accurate Performance Evaluation of Multirate, Multihop Communication Networks. *IEEE/ACM Trans. Networking*, 5(2):266–277, 1997.

[7] N. B. Bean, R. J. Gibbens, and S. Zachary. Asymptotic Analysis of Single Resource Loss Systems in Heavy Traffic, with Applications to Integrated Networks. Preprint.

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

[9] J. S. Kaufman. Blocking in a Shared Resource Enviornment. *IEEE Trans. Commun.*, 29(8):1474–1481, 1981.

[10] N. B. Bean, R. J. Gibbens, and S. Zachary. The Performance of Single Resource Loss Systems in Multiservice Networks. Preprint.

[11] D. Bertsekas and R. Gallager. *Data Networks.* Prentice Hall, 2nd edition, 1994.

[12] F. P. Kelly. Blocking Probabilities in large Circuit Switched Networks. *Adv. Appl. Prob.*, 18:473–505, 1986.