# Fixed Point Approximation for Multirate Multihop Loss Networks With State-Dependent Routing

Mingyan Liu, *Member, IEEE,* and John S. Baras, *Fellow, IEEE*

***Abstract*—In this paper we consider a class of loss networks that have arbitrary topologies and routes of arbitrary length. Multiple traffic classes are present, each with different bandwidth requirement, and each routed according to a state-dependent routing scheme. In particular, we consider the *least loaded routing* method generalized to routes of arbitrary number of hops. The connection level performance metric of interest is the end-to-end blocking probability. We are interested in developing fast evaluation methods to provide reasonably accurate estimates of the blocking probability, especially under heavy traffic load. Our algorithms are based on the fixed-point method framework, also known as the reduced load approximation. In addition to what commonly examined by previous work, two more factors contribute to the complexity of the computation in the scenario under consideration in this paper. One is the state-dependent nature of the routing mechanism, the other is the possible overlapping between routes due to the general multihop topology of the network. We present two fast approximation algorithms to evaluate the blocking probability with state-dependent routing by simplifying the route overlapping computation. We discuss the computational complexity of our algorithms as well as sources of approximation error. We then compare the numerical results with that of simulation and show that our algorithms provide fairly accurate blocking probability estimates especially under heavy traffic load.**

***Index Terms*—Blocking probability, fixed point approximation, least loaded routing, loss network, multihop, multirate, performance modeling, state-dependent routing.**

## I. Introduction

In this paper we study the problem of evaluating connection level blocking probabilities in a class of loss networks with arbitrary topology and state-dependent routing. The focus is on developing fast computational evaluation methods. In a loss network traffic arrives in the form of *calls* or connections, each requiring a fixed amount of bandwidth on every link along a path/route chosen between the source and destination nodes. Upon a call arrival, if the network has a route with the required bandwidth available on all its links, the call is admitted and set up, and it will be holding the requested bandwidth for the entire duration of the call; otherwise the call is rejected or *blocked*. Upon the departure of a call, the occupied bandwidth is released from all the links of that route. The connection or call blocking probability associated with such a loss network is the probability that a call finds the network unavailable when it arrives and is thus rejected. An excellent detailed discussion of loss network can be found in [1].

The concept of loss network is a good abstraction for circuit switched networks, e.g., telephone networks, where each telephone call requires a fixed amount of bandwidth and only releases it upon completion. When the network becomes overly crowded a call may not go through. An important performance measure in the design of such networks is the call blocking probability, and it has been extensively studied for traditional telephone/voice networks. An ATM network can also be viewed as a loss network at the connection level. More generally a data network with certain notion of connection establishment may be considered a loss network if each connection requires guaranteed resources from the network in order to achieve some quality of service (QoS) guarantee, e.g., via the use of certain QoS routing schemes [2], [3]. These connections may be rejected if the network does not have the requested resources. The connection blocking probabilities in such networks can be calculated by applying the concept of effective bandwidth [4]; see, for example, [5], [6]. More recently the loss network model has also been applied to the emerging optical networks.

Analytical methods of evaluating the connection blocking probability are attractive not only because they can potentially generate estimates orders of magnitude faster than simulation, but also because they can be used in network sensitivity analysis, network design and optimization [7], [8]. The Erlang formula

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

established the loss probability of a single link with $C$ units of bandwidth where calls arrive as a Poisson process with rate $\nu$. The blocking probability when multiple classes of calls are present over a single link has also been studied; see, for example, [9], [10].

Analytically, when there are multiple links and multiple classes of calls with different arrival rates and different bandwidth requirement, and when a fixed route is associated with each source-destination node pair, a loss network can be modeled as a multi-dimensional Markov process, with the dimension of the state space being the product of the number of routes allowed in the network and the number of call classes [1]. This is because the number of calls of each class on each feasible route uniquely defines the state of the network. This

Manuscript received March 25, 1999; revised December 13, 2000, and February 17, 2003. This work was supported by the Center for Satellite and Hybrid Communication Networks, under NASA Cooperative Agreement NCC3-528.

M. Liu is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: mingyan@eecs.umich.edu).

J. S. Baras is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: baras@isr.umd.edu).

Digital Object Identifier 10.1109/TNET.2004.826248

Markov process possesses a product form which simplifies the computation of the solution. When alternative routes are present, i.e., when each source-destination node pair is allowed more than one route, the Markov process no longer has a product form. The equilibrium state probabilities can be obtained by writing out the entire set of detailed balance equations and solving them [1]. This approach however, is not practical in dealing with large networks with a large number of routes and integrated services with potentially a large number of service classes, 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 fast computational techniques that provide accurate estimates. In this study we will concentrate on developing such approximation schemes for networks with multiple classes of traffic (i.e., multirate) and multiple routes between any source-destination node pair. For the rest of our discussion we will use the terms *calls* and *traffic* interchangeably.

Blocking probabilities in a loss network, and the *reduced load approximation* (also known as the *fixed point method*) proposed for computing blocking probabilities have been studied extensively; see, for example, [1], [6], [11]–[22]. The reduced load approximation is based on the following two assumptions.

1) *Link independence assumption*. Under this assumption, blocking is regarded as to occur independently from link to link. This assumption allows us to compute the blocking probability at each link separately.
2) *Poisson assumption*. Under this assumption, calls arrive at a link as a Poisson process and the corresponding arrival rate is the original external offered rate thinned by blocking on other links, thus known as the *reduced load*.

Consider the case of a single class of calls with fixed routing. Using Erlang's formula, the blocking probability of each link can be expressed by the offered call arrival 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 the blocking probability of each link. Recent work on using reduced load approximation for fixed routing can be found in [12], [14], [21], [23], and [24].

The reduced load approximation has also been used for sequential alternative routing [20], dynamic alternative routing [24], and state-dependent routing [13], [15], [16].

An important type of state-dependent routing is the *least loaded routing* (LLR), where if a call cannot be set up along the direct route (assumed to exist), then the two-hop alternative route (also assumed to exist) with the largest amount of point-to-point free bandwidth is selected. LLR has been studied in a fully connected symmetric network by Mitra, Gibbens, and Huang in [15] and [16]. Chung, Kashper, and Ross studied LLR for fully connected asymmetric networks in [13]. Girard and Bell further studied LLR for nonfully connected networks (but assuming the direct route and two-hop routes exist) in [22] and [25].

In this study, we are interested in networks that are much sparser than fully connected. Thus, the assumption of the existence of a direct route or two-hop routes between source and destination nodes does not hold in general. In such networks routes can consist of a much larger number of hops and there are typically a large number of possible routes between source and destination nodes. One of the difficulties that arises with such general networks is the dependence or overlapping among routes (via shared common links) employed by a source-destination node pair, as we discuss in more detail in subsequent sections. Such dependence does not exist in a fully connected network where only routes up to two hops are considered.

We will study a generalized version of LLR applied to an arbitrary topology multihop network, where the route with the maximum amount of free/residual bandwidth on its most congested link is chosen for an incoming call. We present two algorithms to estimate call blocking probabilities with LLR in an arbitrary topology network with multiple classes of traffic. In the first algorithm the overlapping among alternative routes is partially ignored to simplify computation. In the second algorithm the overlapping is dealt with by considering the most congested link (determined by the steady state link occupancy distribution) along that route. Both are approximations designed with the intention of producing reasonable and fast estimates. Both have varying degrees of accuracy depending on the network structure as well as the intensity of traffic, examined through comparison with simulation.

Among the aforementioned previous work, [13], [20], and [22] are most relevant to our study here. Greenberg and Srikant in [20] considered a sequential routing scheme in a multirate random topology network, where two nodes may or may not be directly connected. Each node pair is assigned an ordered list of feasible routes in increasing length. Routing is accomplished by going through this list till an admissible route is found. If none is found, the call is blocked. Therefore, a route is chosen only when all routes listed before this one are not in a state to admit the call. A fixed point approximation method was proposed to calculate the blocking probabilities in this scenario. In order to compute the reduced load, the probability of choosing a particular route needs to be derived, and it was done by solving a network reliability problem via a linear program. This approach successfully produced probabilities of using a particular route while taking into account the overlapping among routes on the ordered list. Since we also consider a multihop random topology network, similar route overlapping exists in our study. However, the state-dependent routing scheme considered in this paper is different from the sequential routing considered in [20] and thus different approximation is needed to derive the probability of choosing a route.

Chung, Kashper, and Ross in [13] considered LLR in a fully connected network, and Girard and Bell in [22] considered LLR in a nonfully connected network but with direct and two-hop routes between any node pair. The LLR considered in our study can be viewed as a generalization of the LLR used in these two papers in general multihop network where direct route and two-hop routes do not necessary exist. Due to this generalization, overlapping between different routes used by the same source-destination node pair is introduced and complicates the computation. Similar to [22] we model LLR using a stationary routing scheme where each route is associated with a probability of being chosen. However, the derivation of this probability is different in our paper. In one of our algorithms this probability is computed in a way that can be viewed as a generalization of that used in [13].

The rest of this paper is organized as follows. In Section II we describe the network model, the routing scheme considered, and a summary of assumptions and notations. Section III presents the proposed fixed point approximations, and Section IV compares the approximation with simulation and discusses the accuracy of our models. Section V concludes the paper.

## II. Network Model

Consider a network that has a set of $N$ nodes and a set of $J$ links. The topology of the network is arbitrary, with no special features (we will also call this a random topology, to be distinguished from a mesh topology). In particular, there may or may not be a direct link between a potential source-destination node pair. Each link $j, j = 1, 2, \ldots, J$, has a capacity of $C_j$ in units of bandwidth (interchangeably used with *circuits* and *trunks*). The network supports $S$ classes of traffic or calls. Calls of different classes have different characteristics and resource requirement, e.g., arrival rate, call duration, and bandwidth requirement. In this study the admission of a call is solely determined by the availability of bandwidth. A call of class $s$, $s = 1, 2, \ldots, S$, is accepted into the network if the bandwidth requirement of the call can be accommodated, and is blocked otherwise. If the network does not employ any form of admission control, then a call can be accommodated if there is a feasible route for the call where all links on that route have enough free bandwidth to accept the call. If call admission control is employed, then there may be additional requirements on these links to be met before the call can be admitted. Once a call is accepted onto a certain route, it holds the amount of required bandwidth on all the links along that route for the duration of the call, and releases it upon departure.

The performance metric of interest is the probability that a call of a certain class is blocked. The evaluation of this measure depends on, among many factors, the routing mechanism, the admission control method used, as well as the structure of the network. In the following we will discuss the state-dependent routing mechanism studied in this paper and summarize notations and key assumptions underlying our models.

### A. State-Dependent Routing

State-dependent routing is a commonly studied routing policy, under which a call is assigned to a certain route based on the state of the network, e.g., link congestion level. One important scheme of this kind is the least loaded routing (LLR). Most of previous studies on LLR have concentrated on networks with routes of up to two hops, e.g., [13], [22]. In this scenario, a call is first tried on the direct route, if there is one. If it cannot be setup along the direct route, then the two-hop 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 [13].

A natural extension of LLR to networks where routes tend to have a larger number of hops than one or two hops is to choose the route that has the maximum units of end-to-end free bandwidth (also called the *residual bandwidth*) among all routes. More specifically, in this study we will assume that each source-destination node pair is allowed a list of feasible routes, ordered in increasing length, i.e., number of hops. A call is then routed on the one that has the largest amount of end-to-end residual bandwidth. If multiple routes have the same amount of residual bandwidth the shortest one will be chosen, with ties broken randomly. This essentially results in a "widest first" routing scheme. It is also a "max-min" type of routing since the chosen route has the maximum amount of free bandwidth on its most congested link compared to other alternative routes. One of the approximation algorithms presented in this paper is established by exploiting this max-min property of this routing scheme. In the version of LLR studied in this paper we will not require that the direct link (one-hop route) always be selected with priority over all other routes, but rather that it is selected if it has the maximum residual bandwidth. This is because in a general network direct-link routes do not always exist. We do study a fully connected network in Section IV where we compare the performance of this generalization and the traditional "direct-first" approach. It is worth mentioning that the sequential routing studied in [20] where routes are ordered in increasing length is a "shortest first" routing scheme.

This maximum residual bandwidth routing scheme tries to avoid bottlenecks on a route. However, since a route is chosen only based on the amount of free bandwidth, we may be forced to take a longer or even the longest route in the feasible route set, using more network resources. This may in turn force calls arriving later to also be routed on their longer/longest routes, which leads to increased loss/blocking probability in a network [1]. Therefore, using some form of admission control along with this routing scheme is a valid choice when traffic is heavy. The type of call admission control considered in our model is known as *trunk reservation*, where a call is rejected if the amount of free bandwidth along a route is below a certain level. More specifically, a call of class $s$ is accepted on a route only if every link on this route has at least $b_s + w_s$ units of free bandwidth, where $b_s$ is the bandwidth requirement of class $s$ calls and $w_s$ is the trunk reservation parameter of class $s$. Thus, $w_s$ should be viewed as the amount of bandwidth reserved for calls *other than* class $s$. Equivalently, denoting by $n_i$ the number of ongoing class $i$ calls on a link, a class $s$ call is accepted to a route only if for all links $j$ on the route, we have

$$b_s \leq C_j - \sum_i b_i n_i - w_s.$$

In networks with routes of at most two hops, trunk reservation is typically applied only to the two-hop routes. This means that unless a two-hop route has the extra bandwidth in addition to what is required by a call, the call cannot be admitted. This gives the single-hop route clear priority. Since a single-hop route involves less network resource than a two-hop route, this admission control subsequently makes the admission of a call onto its two-hop routes less likely to cause a later call to be blocked out of its single-hop route. However, it is less straightforward how this reservation scheme should be used in a more general network of an arbitrary topology where routes can be much longer than two hops. One could apply an increasing reservation parameter $w_s$ to routes of increasing length, with $w_s = 0$ for a single-hop route. More generally one could consider a routing scheme that selects a route with least *cost*, which is a function of both the route length and the amount of residual bandwidth along the route, i.e., a mixture of widest first and shortest first routing schemes. However, this is out of the scope of this paper, and in our numerical evaluation in Section IV we will only use trunk reservation in a fully connected network example with one and two-hop routes.

### B. Assumptions

Key assumptions adopted in this paper are as follows, including those that underly the general reduced load approximation technique.

(A1) All links are assumed to be undirected. For traffic between two nodes, we will not differentiate the source from the destination. Consequently a feasible route set is associated with a pair of nodes, regardless of the ordering. This assumption is adopted only for the simplicity of notation and our discussion. Our models can be applied to directional link scenarios in a straightforward manner.

(A2) Calls arrive at the network as a Poisson process, and the total offered load to an individual link is also a Poisson process with rate thinned by blocking on other links.

(A3) Blocking occurs independently from link to link, determined by their respective arrival rates. That is, even though the conditions of successive links along a route are dependent (so is the blocking on these links), we will nevertheless treat them as being independent. This assumption becomes more reasonable as traffic gets heavier.

(A4) We will assume that given stationary inputs, certain time varying quantities of interest have well-defined averages. These include the number of on-going calls on a link of each class, the average call holding time, and the reduced load (call arrival rate) on a link. With these averages we can further assume that there is a stationary probability of choosing a particular route under the state-dependent routing scheme. Thus, the key is to find these probabilities so that the state-dependent routing can be approximated with a stationary, nonstate-dependent routing algorithm with the derived probabilities of route selection. This is discussed in more detail in the next section when our approximation is presented.

### C. Notations

We summarize below the notations used in the paper.

$N$ — The set of nodes in the network. We will use $N$ to denote both the set and the total number of nodes without causing ambiguity.

$J$ — The set of links in the network. Again we will use $J$ to denote both the set and the total number of links in the network.

$C_j$ — The capacity/total bandwidth of link $j$, in units of bandwidth, circuits, or trunks.

$\hat{C}_j$ — The amount of residual/free bandwidth on link $j$. This is a random variable since it is a function of the state of the network.

$R$ — Both the set and the total number of node pairs in the network. Since we ignore the ordering of a pair, $R = N(N-1)/2$.

$M_r$ — The set of routes allowed between node pair $r$. We will also use $M_r$ to denote the total number of routes between node pair $r$.

$r_m$ — The $m$th route of the source-destination node pair $r$. Here $m = 1, 2, \ldots, M_r$. $r_m$ defines a set of links.

$S$ — The total number of traffic/call classes. Each class $s$ has a bandwidth requirement denoted by $b_s$, and a mean call holding time denoted by $\mu_s$. Note that for different node pairs the classification of calls does not have to be the same. However, we will restrict our discussion to a single unified classification since it can always be obtained by increasing the number of classes.

$\lambda_{rs}$ — The arrival rate of class $s$ calls between node pair $r$.

$B_{rs}$ — The end-to-end blocking probability of a class s call between node pair $r$.

$\nu_{js}$ — The reduced load or 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, or the admissibility probability of link $j$.

$p_j(n)$ — The stationary occupancy probability of link $j$, i.e., the probability that exactly $n$ circuits/trunks are being used on link $j$. $n = 0, 1, \ldots, C_j$.

$q_{rs}^m$ — The probability that the $m$th route $r_m$ is chosen for a class $s$ call request between node pair $r$.

$\nu_{js}^{r_m}$ — The reduced load on link $j$ contributed to by class $s$ traffic routed on $r_m$.

$L(r_m)$ — The most congested link along route $r_m$ defined as the link that has the least amount of free bandwidth along a route. That is, $L(r_m) = \text{argmin}_{j \in r_m} \hat{C}_j$. This is in general a random variable since it is a function of $\hat{C}_j$, and thus a function of the state of the network.

$L(r_m - r_k)$ — The most congested of all links that are on route $r_m$ but not on route $r_k$.

$C(r_m)$ — The capacity of the most congested link along route $r_m$, i.e., $C(r_m) = C_{L(r_m)}$.

$\hat{C}(r_m)$ — The residual bandwidth on the most congested link along route $r_m$, i.e., $\hat{C}(r_m) = \hat{C}_{L(r_m)} = \min_{j \in r_m} \hat{C}_j$.

$C_{min}(r_m)$ — The minimum link capacity along route $r_m$, i.e., $C_{min}(r_m) = \min_{j \in r_m} C_j$.

As a general rule, we will use subscripts for links, node pairs, and call classes, while using superscripts to indicate a particular route.

## III. Fixed Point Models

Before we proceed with the fixed point model, it helps to take a closer look at what constitutes the end-to-end blocking probability of a call when a state-dependent routing scheme is used.

### A. Blocking Probability

Consider a random probability based routing policy first. Suppose we associate an independent probability with each of the routes in the feasible route set of node pair $r$, i.e., we select the $m$th route $r_m$ for an incoming call of class $s$ with a fixed probability $p_{rs}^m$ that does *not* depend on link state, with $\sum_m p_{rs}^m = 1$. If the call cannot be admitted by the randomly selected route then it is considered blocked. Under such a routing policy, the end-to-end blocking probability is

$$B_{rs} = 1 - \sum_m p_{rs}^m \prod_{j \in r_m} a_{js}. \tag{1}$$

That is, the probability a call is admitted is the sum of individual route admissibility probability weighted by the probability the route is chosen. Note that (1) is obtained because the route selection probability $p_{rs}^m$ is independent of link admissibility probability $a_{js}$. With this type of routing, the reduced load on a link is the offered load first weighted by the probability of using a route that contains that link and then thinned by blocking on other links on that route.

Consider now the state-dependent routing LLR described in the previous section. Under this scheme each route is *not* chosen independently of link state. The end-to-end blocking probability is, therefore

$$B_{rs} = 1 - \sum_m Pr\,[r_m \text{ is the least loaded route, and } r_m \text{ is in a state to admit the call}]. \tag{2}$$

Note that the elements within the summation in (2) are mutually exclusive (assuming a lower numbered or shorter route is considered to be less loaded among multiple routes that happen to have the same residual bandwidth). However, the event that route $r_m$ is the least loaded of all $r$'s routes and the event that $r_m$ is in a state of admitting class $s$ calls are generally not independent, since they are both functions of the link state. Therefore, this joint probability cannot in general be separated into two individual probabilities.

One way to compute this joint probability is to ignore the dependence and assume that the probability of selecting a route is independent of link state by using assumption (A4), and these probabilities are given by $q_{rs}^m$. Then we can use (1) to approximate (2). This is equivalent to assuming that there exists an average probability that route $r_m$ is the least loaded of all routes. In essence, we are approximating the state-dependent routing by a nonstate-dependent routing, that simply assigns fixed probabilities to each route. In the former traffic is routed based on link state, a scheme also known as *metering*. In the latter, traffic is routed randomly among routes with fixed probabilities, a scheme also known as *randomization*. In general metering results in better performance than randomization; see, for example, in the case of a queueing system [26]. We will use this approximation in our reduced load models.

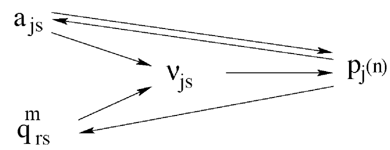


Fig. 1. Mappings between variables.

In order to compute $B_{rs}$, in the rest of this section we show how to find $q_{rs}^m$, the route probability, and $a_{js}$, the link admissibility probability.

### B. Mappings

Our fixed point approximation model centers around a set of mappings between unknowns $\nu_{js}$, $a_{js}$, $p_j(n)$ and $q_{rs}^m$.

First, we fix the link admissibility probability $a_{js}$ and route probability $q_{rs}^m$ to get $\nu_{js}$, the reduced load on link $j$. Then we fix $\nu_{js}$ to get the link occupancy probability $p_j(n)$ and $a_{js}$. Finally we fix $p_j(n)$ to get $q_{rs}^m$. By repeated substitution, the equilibrium fixed point can be solved for all four sets of unknowns. The mappings are illustrated in Fig. 1.

*1) Mapping 1 ($a_{js}$, $q_{rs}^m \longrightarrow \nu_{js}$):* Recall that $\nu_{js}^{r_m}$ is the offered reduced load on link $j$ contributed by traffic class $s$ on route $r_m$ and thinned by blocking on other links. It is given by the reduced load approximation as

$$\nu_{js}^{r_m} = \lambda_{rs} q_{rs}^m I[j \in r_m] \prod_{i \in r_m, i \neq j} a_{is} \tag{3}$$

where $I$ is the indicator function. Note that we first take a portion of the total offered load $\lambda_{rs}$ that is routed on $r_m$ with probability $q_{rs}^m$, and then multiply it with the probability that this portion is admitted by all links other than link $j$. The aggregated load of class $s$ calls on link $j$ is

$$\nu_{js} = \sum_{r \in R} \sum_{r_m \in M_r} \nu_{js}^{r_m}. \tag{4}$$

*2) Mapping 2 ($\nu_{js} \longrightarrow a_{js}, p_j(n)$):* Given $\nu_{js}$, 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 [27] when there is no admission control, or using approaches proposed by Bean, Gibbens and Zachary in [9] and [10]. By the link independence assumption, this mapping is conducted on a per-link basis. Each link is calculated separately and similarly.

In the absence of admission control, classical product form holds for describing the equilibrium call blocking probabilities for a single link [1]. In [27] Kaufman gives a simple one-dimensional recursion for calculating the link occupancy probabilities. Denoting by $n_s$ the number of class $s$ calls in progress on link $j$ and denoting by $b_s$ the bandwidth requirement of class $s$ calls, we have

$$np_j(n) = \sum_{s \in S} b_s \frac{\nu_{js}}{\mu_s} p_j(n - b_{js}), \quad n = 1, \ldots, C_j \tag{5}$$

where $n = \sum_s b_s n_s$. Note that $p_j(n) = 0$ if $n < 0$ and $\sum_{n=0}^{C_j} p_j(n) = 1$.

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

$$a_{js} = 1 - \sum_{n=C_j-b_s+1}^{C_j} p_j(n) = \sum_{n=0}^{C_j-b_s} p_j(n). \quad (6)$$

From (5) we see the computational cost in this case is both linear in $C$ and linear in $S$.

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 the equilibrium distribution of the associated Markov chain, whose state space is a lattice embedded in the simplex $\sum_s b_s n_s \le C, n_s \ge 0$. The dimension of this Markov chain is both proportional to $S$ and $C$, thus the computational cost can be prohibitive, even for moderate values of $C$ and $S$, as pointed out by [20]. An approximation approach was proposed in [9] and [10] that transforms the above problem into a one-dimensional one by assuming that while $n_s$, the number of class $s$ calls in progress varies, the proportion of such calls in progress remains fixed (or varies slowly). In our model we will adopt the following method used in [20], based on the approximation suggested by [9] and [10]. Let $\alpha_{js}$ denote the average number of calls of class $s$ in progress on link $j$, then

$$\alpha_{js} = \frac{a_{js}\nu_{js}}{\mu_s}$$

since calls enter at rate $a_{js}\nu_{js}$ and depart at an average aggregate rate of $\alpha_{js}\mu_s$.

Consider the one-dimensional Markov chain, for any given state $n$ and class $s$, with the following state transition rates:

From state $n$ to state $n + b_s$: $\nu_{js} I(C_j - n \ge w_s + b_s)$;

From state $n$ to state $n - b_s$: $\mu_s n(\alpha_{js}/\sum_t \alpha_{jt}) I(n \ge b_s)$.

This mounts to approximating the actual amount of bandwidth occupied by class $s$ on the link using $n(\alpha_{js}/\sum_t \alpha_{jt})$. The probability of admitting a call of class $s$ is given by

$$a_{js} = 1 - \sum_{n=C_j-b_s-w_s+1}^{C_j} p_j(n) = \sum_{n=0}^{C_j-b_s-w_s} p_j(n). \quad (7)$$

Note that $p_j(n) \longrightarrow \alpha_{js} \longrightarrow p_j(n)$ forms another fixed point, which can be solved via iteration to get the equilibrium distribution $p_j(n)$ and $a_{js}$. Since this is a one-dimensional approximation, it has a per iteration cost linear in $C$.

*3) Mapping 3: ($p_j(n) \longrightarrow q_{rs}^m$):* In this subsection we derive $q_{rs}^m$ from link occupancy distribution $p_j(n)$. The idea is that given the link occupancy probabilities of all links, and given the composition of all feasible routes in the network, we can find out the probability that a particular route is the least loaded among a set of routes.

Denote by $A_n(r_m)$ the event that all links on route $r_m$ have at least $n$ free trunks ($\ge n$), and $\bar{A}_n(r_m)$ the event that at least one link on $r_m$ has less than $n$ free trunks ($< n$). Thus

$$\Pr[A_n(r_m)] = \prod_{j \in r_m} \sum_{k=0}^{C_j-n} p_j(k)$$
$$\Pr[\bar{A}_n(r_m)] = 1 - \Pr[A_n(r_m)].$$

Denote by $\tilde{A}_n(r_m)$ the event that all links on $r_m$ have at least $n$ free trunks and that at least one link on $r_m$ has exactly $n$ free trunks. Therefore

$$\Pr\left[\tilde{A}_n(r_m)\right] = \Pr[A_n(r_m)] - \Pr[A_{n+1}(r_m)].$$

Since $r_m$ can be viewed as a set of links, the above definitions readily apply to an arbitrary set of links rather than a route. We will also use the same notations for similar quantities defined for a single link $j$. More specifically, $A_n(j)$ denotes the event that link $j$ has at least $n$ free trunks, and so on. Thus, we have $A_n(j) = \sum_{k=0}^{C_j-n} p_j(k)$, $\bar{A}_n(j) = 1 - A_n(j)$, and $\tilde{A}_n(j) = p_j(n)$.

The probability $q_{rs}^m$ that a call of class $s$ is routed on $r_m$ is the probability that all routes prior to the $m$th route on the ordered route list have less end-to-end free bandwidth, and that all routes following the $m$th route on the list have at most the same amount of free bandwidth (as we mentioned earlier the routing is such that the shortest one is chosen when there is a tie in end-to-end free bandwidth). Therefore, $q_{rs}^m$ is given by

$$q_{rs}^m = \sum_{n=0}^{C_{min}(r_m)} \Pr\left[\bigcap_{k=1}^{k=m-1} \bar{A}_n(r_k) \bigcap_{k=m+1}^{k=M_r} \bar{A}_{n+1}(r_k) | \tilde{A}_n(r_m)\right] \Pr\left[\tilde{A}_n(r_m)\right]. \quad (8)$$

Note that the range of the summation is from $n = 0$ to the minimum link capacity along route $r_m$, $C_{min}(r_m)$. This is because for $n > C_{min}(r_m)$ the second probability inside the summation would be zero. Because of the overlapping among routes, as the number of alternative routes $M_r$ increases there is no simple way of computing the first probability inside the summation. One may adopt the technique developed in [28] as was used in [20] to approximate the probability of the intersection of non-mutually exclusive events via a linear program. However, the required number of such operations is on the order of $(C - b_s)$, where $C$ is the average link bandwidth. This can be significant when links have a large amount of bandwidth while the bandwidth requirement of each call is relatively small.

A straightforward approximation to (8) is to assume that all routes are independent, i.e., no two routes between a source-destination pair share a common link. Under this assumption all events within the above probability become independent, which yields

$$q_{rs}^m = \sum_{n=0}^{C_{min}(r_m)} \prod_{k=1}^{k=m-1} \Pr\left[\bar{A}_n(r_k)\right] \cdot \prod_{k=m+1}^{k=M_r} \Pr\left[\bar{A}_{n+1}(r_k)\right] \cdot \Pr\left[\tilde{A}_n(r_m)\right]. \quad (9)$$

For networks that do not allow routes longer than two hops, there is no overlapping between routes. Thus, (9) is precisely equivalent to (8). This model easily extends to routing schemes that require the direct route to be used whenever possible by simply replacing the first element of the product ($k = 1$) with $\Pr[\bar{A}_{b_s}(r_k)]$, which is the probability that the first direct route is not in a state to accept the call. We will call this routing scheme *direct first* (DF).

For general multihop networks, we can refine (9) by only considering the dependence between $r_m$ and all other routes but ignoring the overlapping among routes other than $r_m$. This simplification leads to the following:

$$q_{rs}^m = \sum_{n=0}^{C_{min}(r_m)} \prod_{k=1}^{k=m-1} \Pr\left[\bar{A}_n(r_k - r_m)\right] \cdot \prod_{k=m+1}^{k=M_r} \Pr\left[\bar{A}_{n+1}(r_k - r_m)\right] \cdot \Pr\left[\tilde{A}_n(r_m)\right] \quad (10)$$

where $r_k - r_m$ denotes all links that belong to $r_k$ but not $r_m$. This is our first approximation algorithm which we call FPA1. Intuitively, this approximation would be reasonably accurate in a network when routes between every source-destination node pair share one or more common links, but are disjoint otherwise. This is because for any route $r_m$, (10) takes into account the overlapping between $r_m$ and other routes but ignores common links among other routes.

We now examine the quantity in (8) by considering the most congested link along a route. Since the most congested link along a route is the link that has the least free bandwidth among all links on that route, the event $\bar{A}_n(r_m)$ that at least one link on $r_m$ has less than $n$ free trunks is equivalent to the event that the most congested link on $r_m$, denoted by $L(r_m)$, has less than $n$ free trunks, denoted by $\bar{A}_n(L(r_m))$. Similarly, the event $\tilde{A}_n(r_m)$ is equivalent to the event that the most congested link on $r_m$ has exactly $n$ free trunks, i.e., $\tilde{A}_n(L(r_m))$. Replacing events in (8) with these equivalent events, we have

$$q_{rs}^m = \sum_{n=0}^{C_{min}(r_m)} \Pr\left[\bigcap_{k=1}^{k=m-1} \bar{A}_n(L(r_k - r_m)) \bigcap_{k=m+1}^{k=M_r} \bar{A}_{n+1}(L(r_k - r_m)) | \tilde{A}_n(L(r_m))\right] \cdot \Pr\left[\tilde{A}_n\left(L(r_m)\right)\right] \quad (11)$$

where $L(r_k - r_m)$ denotes the most congested link of all links that are on $r_k$ but not $r_m$. The overlapping between these single links are much easier to deal with by simply comparing $L(r_k)$, $1 \le k \le M_r$, $k \ne m$ and $L(r_m)$. This is our second approximation which we call FPA2.

In the special case of no overlapping between these links, i.e., $L(r_k) \ne L(r_m)$, all $k \ne m$, $k \le M_r$, (11) is simply

$$q_{rs}^m = \sum_{n=0}^{C_{min}(r_m)} \prod_{k=1}^{k=m-1} \Pr\left[\bar{A}_n\left(L(r_k)\right)\right] \cdot \prod_{k=m+1}^{k=M_r} \Pr\left[\bar{A}_{n+1}\left(L(r_k)\right)\right] \cdot \Pr\left[\tilde{A}_n\left(L(r_m)\right)\right]. \quad (12)$$

It remains to determine the most congested link $L(r_k)$ for each route $r_k$ computationally. Under steady state, the congestion of a link is characterized by the amount of average free bandwidth on the link, given by the following proposition.

*Proposition 1:* The average number of busy circuits/trunks on link $j$, denoted by $E[X_j]$, is given by

$$E[X_j] = \sum_{s=1}^{S} \frac{\nu_{js} b_s}{\mu_s} \cdot a_{js}.$$

*Proof:* Using Kaufman's recursion

$$n \cdot p(n) = \sum_{s=1}^{S} \frac{\nu_{js} b_s}{\mu_s} p(n - b_s)$$

and summing over 0 through $C_j$ on both sides, we have

$$E[X_j] = \sum_{n=1}^{C_j} n \cdot p(n) = \sum_{n=0}^{C_j} \sum_{s=1}^{S} \frac{\nu_{js} b_s}{\mu_s} p(n - b_s) = \sum_{s=1}^{S} \frac{\nu_{js} b_s}{\mu_s} \sum_{n=b_s}^{C_j} p(n - b_s) = \sum_{s=1}^{S} \frac{\nu_{js} b_s}{\mu_s} \cdot a_s.$$

■

Therefore, the average free bandwidth on the link is given by $C_j - \sum_{s=1}^{S} (\nu_{js} b_s / \mu_s) \cdot a_{js}$, and the most congested link is given by

$$L(r_m) = \text{argmin}_{j \in r_m} \left( C_j - \sum_{s=1}^{S} \frac{\nu_{js} b_s}{\mu_s} \cdot a_{js} \right). \quad (13)$$

Note that $q_{rs}^m$ derived as above will have the same value for different classes of traffic, since the probability that a route is the least loaded as perceived by an incoming call has nothing to do with the type of the call if no admission control is used. However, this is not the case when trunk reservation admission control is used. Since different classes of calls have different reservation requirement, this probability may vary, as shown in the next subsection.

To summarize, algorithm FPA2 computes the probability that a route $r_m$ is chosen by first determining the most congested link along each route and then computing the probability that the most congested link on $r_m$ has the most residual bandwidth among all most congested links. With a similar spirit to that of assumption (A4), such an approximation implies the existence of a most congested link (with high probability) along each route in steady state. During actual network operation, different links along a route may take turns being the most congested link due to traffic dynamics, especially under light traffic when very few links are truly congested. In this case, there may be a stationary probability that a given link is the most congested among all links on a route, but we may not be able to define a *single* link as the most congested link in steady state. As the network gets more and more congested, a few links (or a single link) may become the most congested with higher and higher probability (or more and more often), thus making the assumption more reasonable. Therefore, intuitively, this approximation may only be accurate in a network where each route in the network with high probability has a single bottleneck link that determines the end-to-end free bandwidth, either due to the discrepancy in link bandwidth (i.e., some links have much lower link capacity than others) and/or due to heavy traffic. This is because under heavy traffic and using state-dependent routing scheme, the probability of selecting a route and its admissibility probability will be determined with high probability solely by the relative congestion on these bottleneck links. On the other hand, under light traffic there may not always be a single bottleneck link with high probability, and thus this might not be a good approximation.

### C. Trunk Reservation Call Admission Control

If trunk reservation is used, the first route (also the shortest one) on the ordered route list of every node pair is given priority, and all routes other than the first route will require an extra $w_s$ trunks to be reserved on their links when admitting a call. Therefore, applying trunk reservation, the original computation of $q_{rs}^m$ in (8) is modified as follows, for $m \neq 1$:

$$q_{rs}^m = \sum_{n=0}^{C_{min}(r_m)} \Pr\left[\bar{A}_n(r_1) \bigcap_{k=2}^{k=m-1} \bar{A}_{n+w_s}(r_k) \bigcap_{k=m+1}^{k=M_r} \bar{A}_{n+w_s+1}(r_k) | \tilde{A}_{n+w_s}(r_m)\right] \cdot \Pr\left[\tilde{A}_{n+w_s}(r_m)\right]. \quad (14)$$

For $m = 1$, (8) becomes

$$q_{rs}^1 = \sum_{n=0}^{C_{min}(r_1)} \Pr\left[\bar{A}_n(r_1) \bigcap_{k=2}^{k=M_r} \bar{A}_{n+w_s+1}(r_k) | \tilde{A}_n(r_1)\right] \cdot \Pr\left[\tilde{A}_n(r_1)\right]. \quad (15)$$

(12) can be modified in a similar way and is thus not repeated.

### D. Summary of the Model

To summarize, the fixed point approximation consists of the following steps. First, use (3) and (4) to compute the link reduced load for every link. Next, use either (5) or (7) to compute the link occupancy distribution for every link, depending on whether trunk reservation is used or not. Then, use (10) or (11) (or their trunk reservation versions) to compute the probability of choosing a route. These three steps are then iterated using repeated substitution to obtain the fixed point, i.e., solution for all the unknowns $\nu_{js}$, $a_{js}$, $p_j(n)$, and $q_{rs}^m$. Finally, the end-to-end blocking is calculated using (1).

### E. Discussion, Computational Cost, and Approximation Error

While it is easy to show the existence of a fixed point under the proposed fixed point approximation by applying Brouwer's fixed point theorem, the uniqueness of this fixed point depends on various factors. It has been pointed out in [1] that in general when alternative routes are used without admission control in a fully connected network, there can be multiple fixed points. In particular, bi-stability has been observed. This is due to the fact that calls admitted to the two-hop routes use more network resources and may force more calls to be routed through their two-hop routes instead of their direct routes. Thus, the network may enter a bi-stable region where there are two equilibrium points, one stable and one unstable. Less can be said about more general topology networks. Unfortunately, it is not clear from the existing literature how one might ensure the convergence to a particular equilibrium point. In [20] it was argued that if the ratio between hop numbers of any two alternative routes is sufficiently large (e.g., greater than 0.5), then the network resources used by routing a call on different alternative routes do not significantly vary, and thus the blocking probability will increase more smoothly with the increase in traffic without going into a bi-stable region. It was also suggested that for many cases of practical interest, these approximations are applicable and are expected to find the equilibrium point (with certain error). In all our numerical experiments presented in the next section our fixed point algorithms converged, although we did observe oscillation in intermediate values before it converged in some cases. More is discussed on this in the next section.

Since the computational cost of our approximation largely relies on the number of iterations required to compute the set of unknowns with desired accuracy, we are only able to discuss the computational cost of each iteration, and will leave the observation from experiments to the next section. The first mapping involves $O(J \cdot S)$ operations of (4), each of which has $O(R \cdot M)$ operations of (3), where $R$ is the number of node pairs and $M$ is the average number of routes each node pair has. The cost of (3) is also linear in the average length in hops of a route, denoted by $H$. The second mapping involves $O(J)$ operations of either the Kaufman recursion or the one-dimensional approximation by Bean, Gibbens and Zachary. They both have a cost of $O(C \cdot S)$. The third mapping involves $O(R \cdot M)$ operations in the case of no admission control and $O(R \cdot M \cdot S)$ operations in the case of admission control.

Below, we compare the cost of computing a single $q_{rs}^m$ value using FPA1 and FPA2. The evaluation of $A_n(r_m)$ for a route $r_m$ involves $O(H)$ operations (multiplication). Under FPA1, each route on the route list is evaluated for every value $n$, which gives $O(M \cdot C)$ such operations, $C$ being the average link capacity. This results in a total $O(M \cdot C \cdot H)$ operations. FPA2 on the other hand evaluates $A_n(j)$ for one link per route. Thus, (12) involves $O(M \cdot C)$ operations. Therefore, FPA2 results in faster per iteration computation than FPA1 as $H$ increases (to be expected when the network becomes large).

The accuracy of our approximation models relies firstly on the validity of the sequence of assumptions made in the previous section, i.e., the independence and Poisson assumptions as well as the existence of well-defined averages for all the values we computed. In general, the independence and Poisson assumptions become more accurate when the network is better connected, routes are diverse and as the traffic becomes heavier. In addition to that, the accuracy heavily relies on the structure of the network. This applies to both FPA1 and FPA2, but the sources of error are not completely the same. FPA1 largely ignores the dependence between routes. Therefore, if we consider the case where a network has mostly disjoint routes/paths and a second case where a network has many routes sharing links, the algorithm will in general produce better approximation in the first case. If routes are not all disjoint but the majority of routes between a given node pair share the same set of links and are otherwise disjoint, then the approximation error may also be reduced. FPA2 approximates the route selection probability by only comparing the most congested links on different routes. Furthermore, these most congested links are defined by their expected occupancies, i.e., these are most congested links in an average sense. In a network with equal link bandwidth and well balanced traffic load and routing choices, the most congested link along a route may not be a fixed link with high probability over time, especially when traffic is light. Therefore, this approximation might only be reasonably accurate when there is a clear single bottleneck link along each path that rarely changes from time to time, i.e., when there is a link that is the most congested with high probability or for a large portion of the time. Unfortunately, the actual condition of a network is usually a mixture of the above scenarios, therefore rendering the assessment of degree of accuracy less straightforward. These errors will be examined through a sequence of numerical examples in the next section. What we hope to gain is fast computation methods by tolerating certain amount of error.

It is worth mentioning that the method underlying FPA2 can be easily extended to other routing schemes. For example, we could use a mixed widest first and shortest first routing scheme by considering both the maximum end-to-end residual bandwidth and the length of a route via the following cost function of route $r_m$:

$$c_1 \sum_{j \in r_m} b_s - c_2 \hat{C}(r_m)$$

where the first term is the total number of trunks that would be occupied if the route is chosen, and the second is the number of free trunks on its bottleneck link (end-to-end free bandwidth), each weighted by constants $c_1$ and $c_2$, respectively. The route which minimizes this cost is chosen. Clearly, a longer route will increase the cost. If $c_1$ is zero, this becomes the widest first LLR we used earlier.

Under this mixed routing scheme, selecting the $m$th route for routing the call indicates that all the routes before the $m$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. Therefore, the probability of attempting the call on the $m$th route can be expressed as

$$q_{rs}^m = \sum_{n=0}^{C_{min}(r_m)} \prod_{k=1}^{m-1} t_{r_k}\left( \frac{c_1}{c_2} \sum_{j \in r_k} b_s - \frac{c_1}{c_2} \sum_{j \in r_m} b_s + n \right) \cdot \prod_{k=m+1}^{M_r} t_{r_k}\left( \frac{c_1}{c_2} \sum_{j \in r_k} b_s - \frac{c_1}{c_2} \sum_{j \in r_m} b_s + n + 1 \right) \cdot \Pr\left[ \tilde{A}_n \left( L(r_m) \right) \right]$$

where $t_{r_m}(n) = \Pr[\bar{A}_n(L(r_m))]$, the probability that the most congested link on route $r_m$ has less than $n$ free trunks.

## IV. Experiment and Evaluation

In this section we give two network examples, comparing the results from the two approximation algorithms (FPA1 and FPA2) to that of simulation. First we study a fully connected network allowing routes up to two hops, with no route overlapping. We then study a general topology network, allowing routes up to a certain number hops. Route overlapping is obvious in this case, and the level of overlapping varies from one node pair to another. Some node pairs have routes sharing a single bottleneck link while others do not.

The stopping criteria for the iterative algorithms is for the difference between successive iterates to be less than 0.001. Simulations are run to achieve 95% confidence intervals with width of 0.001.

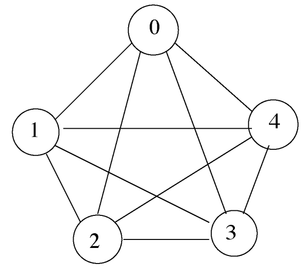


Fig. 2. Example one: fully connected network.

### A. Fully Connected Network

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

We assume each link has a capacity of 100 units of bandwidth or trunks. On the other hand, the offered traffic is asymmetric. 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 increasing order of the numerical index of the intermediate node on the two-hop routes (e.g., route 1-2-4 is listed ahead of 1-3-4). There are three classes of calls/connections indexed by 1, 2, and 3, with bandwidth requirement of 1, 2, and 3 trunks, respectively. When call admission control is used, the trunk reservation parameter for each class is 2, 4, and 6, respectively. That is, the first class will require that a two-hop route to maintain an extra 2 trunks of free bandwidth, and so on. In the Appendix we give a list of nominal traffic rates, and we examine traffic load of 1.2, 1.4, 1.8 and 2.0 times of the nominal traffic, denoted by 1.2x, 1.4x, 1.8x, and 2.0x, respectively. The nominal traffic approximately corresponds to, for every node, a ratio of 0.5 between the total bandwidth requirement *originating from* that node ($\sum_s \lambda_{rs} b_s / \mu_s$, where this node is part of the pair $r$) and the total link capacity attached to that node ($\sum_j C_j$ where $j$ is a link of this node).

The results of two approximation algorithms (FPA1 and FPA2) and the discrete event simulation (DES) for selected node pairs are shown in Table I–IV. "DES(1)" denotes the simulation result for class 1 calls, and "FPA1(1)" denotes the result using FPA1 for class 1 calls, and so on. All blocking probability values are in percentages.

First note that in this example the end-to-end blocking probabilities of the same class traffic are of very similar values regardless of the actual source and destination node pairs. There are two reasons to this. One is because the network is symmetric in terms of topology and link capacity. The other is the fact that the traffic load is such that the total bandwidth requirement originating from any node is roughly the same (as "nominal" load was explained earlier). Therefore, the same class calls experience similar blocking along their direct link routes and two-hop routes irrespective of its source and destination.

We see that when traffic is very light and the end-to-end blocking probability is (far) below 1%, neither algorithm produces accurate estimates, as shown in Table I, with FPA2 generating overestimates of relative errors around +300%.

Overall FPA1 provides quite accurate estimates except for the 1.2x traffic case. The accuracy improves as traffic load increases

TABLE I
EXAMPLE 1 WITH 1.2 TIMES THE NOMINAL TRAFFIC

| | Node Pair (% blocking) | | | | | | |
|---|---|---|---|---|---|---|---|
| | (0,1) | (0,2) | (0,3) | (1,2) | (1,3) | (1,4) | (2,4) |
| DES(1) | 0.02 | 0.01 | 0.02 | 0.02 | 0.02 | 0.01 | 0.02 |
| FPA1(1) | 0.02 | 0.01 | 0.02 | 0.01 | 0.02 | 0.01 | 0.02 |
| FPA2(1) | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 |
| DES(2) | 0.24 | 0.26 | 0.23 | 0.25 | 0.19 | 0.20 | 0.24 |
| FPA1(2) | 0.32 | 0.31 | 0.29 | 0.26 | 0.31 | 0.29 | 0.31 |
| FPA2(2) | 0.61 | 0.45 | 0.41 | 0.61 | 0.73 | 0.78 | 0.76 |
| DES(3) | 0.84 | 0.88 | 0.80 | 0.83 | 0.77 | 0.77 | 0.82 |
| FPA1(3) | 0.98 | 0.92 | 0.89 | 0.90 | 0.86 | 0.88 | 0.97 |
| FPA2(3) | 1.36 | 1.41 | 1.37 | 1.39 | 1.39 | 1.37 | 1.42 |

TABLE II
EXAMPLE 1 WITH 1.4 TIMES THE NOMINAL TRAFFIC

| | Node Pair (% blocking) | | | | | | |
|---|---|---|---|---|---|---|---|
| | (0,1) | (0,2) | (0,3) | (1,2) | (1,3) | (1,4) | (2,4) |
| DES(1) | 0.42 | 0.42 | 0.43 | 0.45 | 0.44 | 0.32 | 0.41 |
| FPA1(1) | 0.47 | 0.49 | 0.47 | 0.46 | 0.44 | 0.44 | 0.47 |
| FPA2(1) | 1.85 | 1.68 | 1.54 | 1.86 | 1.83 | 1.78 | 1.85 |
| DES(2) | 4.10 | 4.46 | 4.21 | 4.40 | 4.09 | 3.78 | 4.12 |
| FPA1(2) | 4.32 | 4.58 | 4.41 | 4.33 | 4.18 | 4.15 | 4.38 |
| FPA2(2) | 16.44 | 17.03 | 16.44 | 16.69 | 16.56 | 16.55 | 16.42 |
| DES(2) | 11.74 | 12.64 | 12.08 | 12.22 | 11.71 | 12.22 | 11.96 |
| FPA1(3) | 12.93 | 13.63 | 13.19 | 12.98 | 12.58 | 13.18 | 13.09 |
| FPA2(3) | 24.13 | 24.91 | 24.15 | 24.43 | 24.24 | 24.23 | 24.18 |

TABLE III
EXAMPLE 1 WITH 1.8 TIMES THE NOMINAL TRAFFIC

| | Node Pair (% blocking) | | | | | | |
|---|---|---|---|---|---|---|---|
| | (0,1) | (0,2) | (0,3) | (1,2) | (1,3) | (1,4) | (2,4) |
| DES(1) | 2.44 | 2.49 | 2.68 | 2.78 | 2.76 | 2.31 | 2.42 |
| FPA1(1) | 2.94 | 3.13 | 3.08 | 3.04 | 2.99 | 2.84 | 3.01 |
| FPA2(1) | 5.22 | 5.60 | 5.91 | 5.47 | 5.61 | 5.61 | 5.79 |
| DES(2) | 17.38 | 18.67 | 18.31 | 18.95 | 18.20 | 16.57 | 17.79 |
| FPA1(2) | 18.24 | 19.23 | 18.98 | 18.78 | 18.55 | 17.81 | 18.59 |
| FPA2(2) | 19.18 | 19.57 | 20.00 | 19.26 | 19.49 | 19.43 | 19.75 |
| DES(3) | 38.78 | 40.19 | 39.90 | 40.65 | 39.74 | 37.53 | 39.21 |
| FPA1(3) | 39.38 | 41.16 | 40.69 | 40.41 | 39.99 | 38.71 | 39.98 |
| FPA2(3) | 41.96 | 42.80 | 43.20 | 42.29 | 42.56 | 42.40 | 42.83 |

for the same class, and as the blocking increases with bandwidth requirement under the same traffic load. For example, in the case of class 3, the worst case relative error ranges from 60% in Table I to 3% in Table IV for the 2.0x case, while for the same level of 1.8x traffic load the worst case relative error ranges from 15% for class 1, to 8% for class 2, to 5% for class 3 in Table III. The absolute error of FPA1 averages below 1% in blocking probability, with the worst case being 1.7% (node pair (0,4), class 3, with 1.4x traffic). This accuracy is to be expected since in this network there is no route overlapping, therefore FPA1 is a fairly accurate model. The improvement with increasing traffic or with increasing blocking probability is likely due to the fact that as traffic becomes heavier assumptions (A2) and (A3) become more accurate.

On the other hand, FPA2 provides good estimates only under very heavy traffic, and only when the experienced end-to-end blocking is high enough (more than 10%). The worst case relative error for classes 2 and 3 in the 2.0x case are 12% and 5%, respectively, with an average absolute error below 1%. This is also to be expected as discussed in the previous section considering the symmetric nature of this network in terms of link capacity.

TABLE IV
EXAMPLE 1 WITH 2.0 TIMES THE NOMINAL TRAFFIC

| | Node Pair (% blocking) | | | | | | |
|---|---|---|---|---|---|---|---|
| | (0,1) | (0,2) | (0,3) | (1,2) | (1,3) | (1,4) | (2,4) |
| DES(1) | 3.83 | 3.7 | 4.07 | 4.26 | 4.33 | 3.52 | 3.72 |
| FPA1(1) | 4.47 | 4.74 | 4.72 | 4.66 | 4.63 | 4.35 | 4.59 |
| FPA2(1) | 8.46 | 8.43 | 8.02 | 7.95 | 8.02 | 7.70 | 7.80 |
| DES(2) | 23.82 | 25.13 | 24.99 | 25.69 | 24.83 | 22.72 | 24.17 |
| FPA1(2) | 24.47 | 25.73 | 25.57 | 25.37 | 25.24 | 24.05 | 25.01 |
| FPA2(2) | 25.66 | 26.84 | 26.24 | 26.06 | 26.15 | 25.61 | 25.84 |
| DES(3) | 48.18 | 49.70 | 49.45 | 50.40 | 49.74 | 46.72 | 48.59 |
| FPA1(3) | 48.23 | 50.24 | 49.95 | 49.73 | 49.47 | 47.68 | 49.04 |
| FPA2(3) | 49.55 | 50.04 | 49.41 | 49.12 | 49.18 | 48.54 | 48.89 |

TABLE V
EXAMPLE 1 WITH 2.0 TIMES THE NOMINAL TRAFFIC WITH TRUNK RESERVATION

| | Node Pair (% blocking) | | |
|---|---|---|---|
| | (0,3) | (1,2) | (2,4) |
| Class | 1 | 2 | 3 |
| DES-TR | 1.94 | 11.38 | 31.93 |
| FPA1-TR | 1.84 | 11.78 | 32.12 |
| FPA2-TR | 4.66 | 23.47 | 34.06 |

TABLE VI
EXAMPLE 1 WITH 2.0 TIMES THE NOMINAL TRAFFIC WITH DIRECT FIRST LLR

| | Node Pair (% blocking) | | |
|---|---|---|---|
| | (0,3) | (1,2) | (2,4) |
| Class | 1 | 2 | 3 |
| DES-DF | 4.05 | 19.43 | 36.74 |
| FPA1-DF | 3.49 | 20.78 | 37.44 |
| FPA2-DF | 8.22 | 21.32 | 41.59 |

Since all links have equal capacity, when traffic is light there is no obvious single bottleneck along a two-hop route. In particular, each of the two links on a two-hop route may be the more congested one equally likely, and thus the most congested link may alternate between the two. As the traffic becomes heavier, congestion becomes more prominent till eventually the congestion on some links is significant enough to make them the more congested link on certain routes with high probability. Note that although the link capacities are equal, the external traffic load is asymmetric. In addition to this, the accuracy of FPA2 improves with increasing traffic and increasing blocking for the same reasons given above for FPA1. From this result, FPA2 should not be recommended to give estimates for light traffic scenarios where some end-to-end blocking probabilities are well below 10%.

Note that in general both FPA1 and FPA2 are conservative algorithms that produce overestimates, which is desirable (given reasonable accuracy) from performance evaluation and system design point of view. The reason behind this overestimate is the fact that we used a nonstate-dependent routing scheme to approximate a state-dependent routing scheme as discussed in the previous section, although the probabilities of selecting a route is computed based on the stationary link state distribution. As mentioned before, the former is a randomization scheme, which in general results in worse performance than the latter (metering). Our algorithms essentially estimate the blocking probabilities by approximating the original state-dependent routing using a randomized routing, resulting in higher blocking probability estimates.

Table V shows results of using trunk reservation (TR-LLR) under heavy traffic 2.0x (we only show one entry per class of traffic due to the fact that again same class traffic experiences very similar end-to-end blocking probabilities). We see from Table V that trunk reservation very effectively reduces end-to-end blocking for every class of traffic, compared to results from Table IV. Table VI shows results for the *direct first* (DF-LLR) type of state-dependent routing. That is, the direct link between any two nodes is always chosen to route an incoming call if it is in a state to admit a call. Otherwise the least loaded two-hop route is chosen. We only show the 2.0x case since the general observation of the performance of FPA1 and FPA2 when the load increases remains the same as in the case of pure LLR.

We have the same observation as before, i.e., FPA2 is only reasonable when the blocking is high and provides gross overestimates for results that are well below 10%. FPA1 on the other hand provides quite accurate estimates with absolute error averaging below 1%. An interesting observation is that both trunk reservation and direct first result in lower end-to-end blocking, but trunk reservation is far more effective than using direct link first. Note that by using trunk reservation, if a call is blocked out of its direct route (indication of congestion), then it has a good chance of being blocked out of its two-hop routes as well due to reservation requirement. Thus, some calls are either routed on their direct route, or not admitted at all. As a result, later calls have a higher chance of being admitted onto their direct routes. On the other hand using DF-LLR calls will be routed on their two-hop routes if the direct one is not available but a two-hop route is (this is easier to achieve under DF than under TR due to reservation requirement). As a result under DF-LLR more calls will be routed on their two-hop routes than under TR-LLR, occupying more network resources. Consequently trunk reservation produces lower blocking probability, for all classes in this case.

### B. Example Two: Random Topology Network

Our next example is borrowed from [20] with minor changes. The topology is derived from an existing commercial network and is depicted in Fig. 3.

There are 16 nodes and 31 links, with link capacities ranging from 60 to 180 trunks. The detailed link capacities as well as the nominal traffic rates can be found in [20], and are not provided in this paper due to space limit. The traffic in the network consists of four classes, requiring bandwidth of one, two, three, and four trunks, respectively. No admission control is employed in this experiment. In [20] sequential routing was used. Here we use LLR, allowing all routes up to six hops, with the total number of routes between each node pair not exceeding seven. Routes between each node pair are ordered in increasing number of hops, with ties broken randomly.

We list results for selected node pairs and classes in Tables VII–XI, corresponding to the nominal, 1.2, 1.4, 1.6, and 1.8 times the nominal traffic (denoted by 1.0x, 1.2x, 1.4x, and 1.8x), respectively. All blocking probabilities are in percentages.

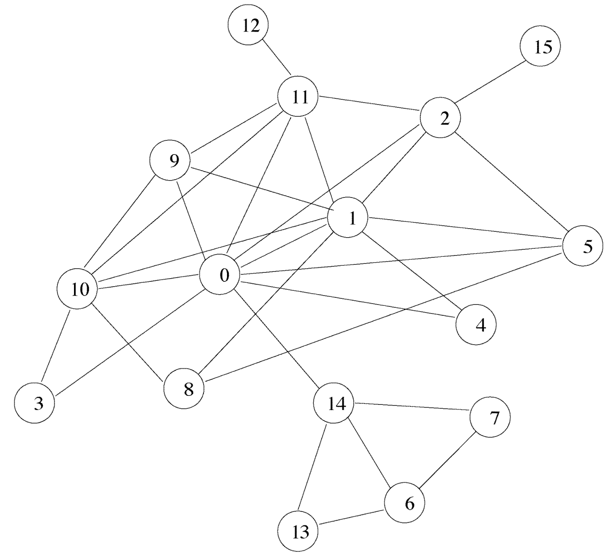


Fig. 3. Example two: a random topology network.

FPA1 and FPA2 again provide overestimates and both algorithms improve as the network becomes more congested. The reason for this is the same as explained in the previous example. That is, heavy traffic makes the underlying assumptions more accurate; and both of our algorithms model a state-dependent routing scheme using a randomized routing approximation. Thus, our results on average are conservative estimates of the actual blocking probabilities. In addition, in the case of FPA2, heavier traffic makes the assumption of a steady state most congested link more accurate.

Similar to the first example, FPA2 only gives reasonable estimates for node pairs that experience significant blocking and when the overall traffic load is high, with average absolute error below 2% in the 1.8x traffic case. However, overall the performance of FPA2 in this example is better than in the previous one. For example, the estimates for node pairs that have blocking probabilities between 5% and 10% have an average relative error around 30%, which is much lower than in the first example (around 100%). The accuracy significantly improves as blocking becomes more severe. The reason FPA2 performs better in this example is due to the asymmetric topology of the network as well as the asymmetry in link capacities, which more easily lead to a dominant bottleneck link along a route, whereas in the previous example this does not happen until the traffic is very heavy.

FPA1 provides quite accurate estimates overall, with an average error slightly higher than in the first example. For example, the absolute error in the 1.8x case averages around 0.7% in blocking probability whereas in the first example the average absolute error in the 1.8x case is slightly below 0.6% in blocking probability. It has to be mentioned that the multiple traffic load cases specified for the two examples are not exactly equivalent considering the network topology, link capacity as well as the different traffic classes. At first thought, FPA1 was not expected to perform very well since there is obvious route overlapping

TABLE VII
EXAMPLE 2 WITH NOMINAL TRAFFIC

| | Node Pair (% blocking) | | | | | | |
|---|---|---|---|---|---|---|---|
| Class | (0,4) 4 | (0,13) 1 | (1,6) 1 | (9,13) 4 | (0,6) 2 | (6,11) 2 | (1,6) 3 |
| DES | 0.00 | 0.75 | 0.79 | 3.62 | 1.55 | 1.09 | 2.23 |
| FPA1 | 0.00 | 0.87 | 0.82 | 3.94 | 2.18 | 1.71 | 2.48 |
| FPA2 | 0.43 | 2.34 | 1.88 | 5.47 | 3.03 | 4.19 | 4.14 |

TABLE VIII
EXAMPLE 2 WITH 1.2 TIMES THE NOMINAL TRAFFIC

| | Node Pair (% blocking) | | | | | | |
|---|---|---|---|---|---|---|---|
| Class | (0,4) 4 | (0,13) 1 | (1,6) 1 | (9,13) 4 | (0,6) 2 | (6,11) 2 | (1,6) 3 |
| DES | 0.00 | 3.64 | 3.65 | 15.23 | 7.16 | 7.07 | 10.39 |
| FPA1 | 0.00 | 3.85 | 4.41 | 15.96 | 8.12 | 7.72 | 11.07 |
| FPA2 | 0.45 | 5.61 | 5.85 | 16.36 | 9.14 | 8.98 | 13.65 |

TABLE IX
EXAMPLE 2 WITH 1.4 TIMES THE NOMINAL TRAFFIC

| | Node Pair (% blocking) | | | | | | |
|---|---|---|---|---|---|---|---|
| Class | (0,4) 4 | (0,13) 1 | (1,6) 1 | (9,13) 4 | (0,6) 2 | (6,11) 2 | (1,6) 3 |
| DES | 0.00 | 7.49 | 7.61 | 28.42 | 14.84 | 14.46 | 21.62 |
| FPA1 | 0.00 | 7.85 | 8.04 | 28.94 | 15.26 | 14.88 | 22.35 |
| FPA2 | 0.54 | 9.39 | 9.81 | 29.74 | 17.58 | 16.17 | 24.67 |

TABLE X
EXAMPLE 2 WITH 1.6 TIMES THE NOMINAL TRAFFIC

| | Node Pair (% blocking) | | | | | | |
|---|---|---|---|---|---|---|---|
| Class | (0,4) 4 | (0,13) 1 | (1,6) 1 | (9,13) 4 | (0,6) 2 | (6,11) 2 | (0,6) 3 |
| DES | 0.00 | 11.85 | 11.80 | 39.90 | 22.49 | 21.40 | 31.38 |
| FPA1 | 0.00 | 11.46 | 12.38 | 40.52 | 23.08 | 21.79 | 31.98 |
| FPA2 | 0.46 | 13.54 | 14.44 | 42.37 | 25.60 | 23.56 | 32.47 |

TABLE XI
EXAMPLE 2 WITH 1.8 TIMES THE NOMINAL TRAFFIC

| | Node Pair (% blocking) | | | | | | |
|---|---|---|---|---|---|---|---|
| Class | (0,4) 4 | (0,13) 1 | (1,6) 1 | (9,13) 4 | (0,6) 2 | (6,11) 2 | (1,6) 3 |
| DES | 0.25 | 14.98 | 15.25 | 49.17 | 28.68 | 28.71 | 39.93 |
| FPA1 | 0.31 | 15.39 | 16.48 | 49.16 | 29.74 | 29.09 | 40.17 |
| FPA2 | 1.05 | 15.78 | 16.42 | 50.13 | 30.11 | 30.13 | 40.36 |

in this example. However a closer inspection of this random topology network reveals some interesting features. Note that although this network is not at all well connected as a whole, it consists of three distinct groups of nodes. The first group consists of nodes 0–5 and 8–9. Note that this group of nodes are very well connected among themselves. For example, between node 0 and node 1 there is one direct route and six two-hop routes that do not overlap. The second group consists of nodes 12 and 15, which are attached to the first group via a single link. Thus, all traffic between either of the two nodes and the rest of the network will share a single link. Similarly the third group, which consists of nodes 6–7 and 13–14, is also attached to the first group via a single link. As a result, most of the node pairs have routes that either do not overlap significantly and/or share common links that are likely to be the bottleneck links. These properties have made the assumptions underlying approximation FPA1 more accurate.

Our results are in general similar to that in [20] (for the entries that were reported in [20]), with no significant performance improvement or degradation (e.g., 14.98% for pair (0,13), class 1, with 1.8x traffic vs. 14.92% in [20], and 39.90% for pair (9,13), class 4, with 1.6x traffic vs. 39.58% in [20]). Indeed our results are slightly above those shown in [20] for the entries that were reported in [20] (total of four pairs). Comparing the shortest first sequential routing used in [20] with the widest first least loaded routing used here, the latter avoids congested links at the possible expense of occupying longer routes and more network resources. It would be interesting to further consider more general combinations of shortest first and widest first routing schemes, such as the cost function introduced at the end of Section III. It has to be mentioned that the two sets of results are not strictly comparable since we have likely adopted different sets of routes (and likely fewer number of routes for each pair) between node pairs. And even in the case where routes are the same, we may have used different ordering of these routes than that used in [20].

### C. Discussion

We have discussed sources of approximation error for both algorithms, and these errors are further examined in the above two examples. We observe that the features of the second network example, i.e., group of very well connected nodes and groups of nodes that share a single bottleneck link, etc., are not at all exclusive to this particular example. They are in fact quite common to many commercial and public networks. Thus, we believe that our algorithms could be applied to a large class of general topology networks.

We did not observe oscillation in the second example. Oscillation was observed while running FPA2 for the 1.2x case of the first example. The algorithm managed to converge via heavy dampening techniques where during the iteration newly computed values are heavily weighted by their old values to prevent drastic changes from happening. As discussed in the previous section, it is likely that this fully connected network without admission control has two equilibrium points.

All simulations and the approximations are run on a Dell Precision 630 workstation. The simulation typically takes on the order of $10^4$ s to complete, while FPA1 typically takes on the order of $10^2$ s and FPA2 takes on the order of 10–$10^2$ s. The number of iterations needed for FPA1 does not change significantly from one experiment to another, while the number of iterations needed for FPA2 typically increases with the increase in traffic intensity. It is a subject of further study whether this is in any way related to the intrinsic structure of these approximation algorithms.

## V. CONCLUSION

In this paper we derived two fixed point approximation algorithms to estimate the connection level blocking probabilities in a general topology network with state-dependent routing schemes. Central to our approximation is the derivation of the probability of choosing a particular route for an incoming call.

While this probability can be expressed in terms of link occupancy probabilities, it is computationally expensive due to overlapping links among routes. Our first approximation is obtained by considering only part of the overlapping, and the second approximation is obtained by considering only the most congested link on a route. We discussed the accuracy, computational cost, and applicability of each of them. An important future research area would be to explore the capabilities of such models in providing performance bounds, and to use such models for solving design problems.

## APPENDIX

TABLE XII
ARRIVAL RATES IN EXAMPLE ONE

| pair $r$ | $s$ | $\lambda_{rs}$ | pair $r$ | $s$ | $\lambda_{rs}$ | pair $r$ | $s$ | $\lambda_{rs}$ |
|---|---|---|---|---|---|---|---|---|
| $(0,1)$ | 1 | 10.0 | $(1,2)$ | 1 | 18.5 | $(2,3)$ | 1 | 0.0 |
| | 2 | 7.5 | | 2 | 10.0 | | 2 | 7.5 |
| | 3 | 5.0 | | 3 | 2.5 | | 3 | 10.0 |
| $(0,2)$ | 1 | 2.5 | $(1,3)$ | 1 | 15.0 | $(2,4)$ | 1 | 7.5 |
| | 2 | 19.0 | | 2 | 3.5 | | 2 | 7.5 |
| | 3 | 4.5 | | 3 | 8.5 | | 3 | 10.0 |
| $(0,3)$ | 1 | 8.0 | $(1,4)$ | 1 | 1.5 | $(3,4)$ | 1 | 25.5 |
| | 2 | 8.5 | | 2 | 10.0 | | 2 | 13.0 |
| | 3 | 8.0 | | 3 | 10.0 | | 3 | 0.0 |
| $(0,4)$ | 1 | 3.0 | | | | | | |
| | 2 | 0.0 | | | | | | |
| | 3 | 16.0 | | | | | | |

## ACKNOWLEDGMENT

The authors would like to thank the associate editor and the anonymous reviewers for their comments in completing the final version of this paper.

## REFERENCES

[1] F. P. Kelly, "Loss networks," *Ann. Appl. Probabil.*, vol. 1, no. 3, pp. 319–378, 1991.

[2] A. Orda, "Routing with end-to-end QoS guarantees in broadband networks," *IEEE/ACM Trans. Networking*, vol. 7, pp. 365–374, June 1999.

[3] R. Guérin and A. Orda, "QoS routing in networks with inaccurate information: theory and algorithms," *IEEE/ACM Trans. Networking*, vol. 7, pp. 350–364, June 1999.

[4] F. P. Kelly, "Notes on effective bandwidth," in *Stochastic Networks Theory and Applications*, F. P. Kelly, S. Zachary, and I. Ziedins, Eds. Oxford, U.K., 1996, pp. 141–168.

[5] J. Y. Hui, "Resource allocation for broadband networks," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 1598–1608, Dec. 1988.

[6] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan, "ATM network design and optimization: a multirate loss network framework," *IEEE/ACM Trans. Networking*, vol. 4, pp. 531–543, Aug. 1996.

[7] A. Girard and B. Liau, "Dimensioning of adaptively routed networks," *IEEE/ACM Trans. Networking*, vol. 1, pp. 460–468, Aug. 1993.

[8] M. Liu, "Network Performance Modeling, Design and Dimensioning Methodologies," Center for Satellite and Hybrid Communication Networks, Univ. Maryland, College Park, Tech. Rep. PhD 2000-6, 2000.

[9] N. G. Bean, R. J. Gibbens, and S. Zachary, "The performance of single resource loss systems in multiservice networks," in *The Fundamental Role of Teletraffic in the Evolution of Telecommunications Networks, Proceedings of the 14th International Teletraffic Congress—ITC 14*, J. Labetoulle and J. W. Roberts, Eds. New York: Elsevier Science B.V., June 1994, vol. 1a, Teletraffic Science and Engineering, pp. 13–21.

[10] N. B. Bean, R. J. Gibbens, and S. Zachary, "Asymptotic analysis of single resource loss systems in heavy traffic, with applications to integrated networks," *Adv. Appl. Probabil.*, pp. 273–292, Mar. 1995.

[11] F. P. Kelly, "Routing in circuit-switched networks: optimization, shadow prices and decentralization," *Adv. Appl. Probabil.*, vol. 20, pp. 112–144, 1988.

[12] ——, "Blocking probabilities in large circuit switched networks," *Adv. Appl. Probabil.*, vol. 18, pp. 473–505, 1986.

[13] S. Chung, A. Kashper, and K. W. Ross, "Computing approximate blocking probabilities for large loss networks with state-dependent routing," *IEEE/ACM Trans. Networking*, vol. 1, pp. 105–115, Feb. 1993.

[14] S. Chung and K. W. Ross, "Reduced load approximations for multirate loss networks," *IEEE Trans. Commun.*, vol. 41, pp. 1222–1231, Aug. 1993.

[15] D. Mitra, R. Gibbens, and B. D. Huang, "State-dependent routing on symmetric loss networks with trunk reservations I," *IEEE Trans. Commun.*, vol. 41, pp. 400–411, Feb. 1993.

[16] D. Mitra and R. Gibbens, "State-dependent routing on symmetric loss networks with trunk reservations II: asymptotics, optimal design," *Ann. Oper. Res.*, vol. 35, no. 1, 1992.

[17] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan, "Virtual private networks: joint resource allocation and routing design," in *Proc. IEEE INFOCOM*, vol. 1, 1999, pp. 480–490.

[18] J. A. Morrison, K. G. Ramakrishnan, and D. Mitra, "Refined asymptotic approximation to loss probabilities and their sensitivities in shared unbuffered resources," *SIAM J. Appl. Math.*, vol. 59, no. 2, pp. 494–513, 1998.

[19] K. R. Krishnan and F. Huebner-Szabo de Bucs, "Admission control and state-dependent routing for multirate circuit switched traffic," in *Proc. 15th Int. Teletraffic Conf.*, Washington, DC, June 1997, pp. 1043–1054.

[20] A. G. Greenberg and R. Srikant, "Computational techniques for accurate performance evaluation of multirate, multihop communication networks," *IEEE/ACM Trans. Networking*, vol. 5, pp. 266–277, Feb. 1997.

[21] W. Whitt, "Blocking when service is required from several facilities simultaneously," *AT&T Tech. J.*, vol. 64, no. 8, pp. 1807–1857, 1985.

[22] A. Girard and M. A. Bell, "Blocking evaluation for networks with residual capacity adaptive routing," *IEEE Trans. Commun.*, vol. 37, pp. 1372–1380, Dec. 1989.

[23] P. J. Hunt, "Implied costs in loss networks," *Adv. Appl. Probabil.*, vol. 21, pp. 661–680, 1989.

[24] F. P. Kelly, "Routing and capacity allocation in networks with trunk reservation," *Math. Oper. Res.*, vol. 15, pp. 771–793, 1990.

[25] A. Girard, *Routing and Dimensioning in Circuit-Switched Networks*, 1st ed. Reading, MA: Addison-Wesley, 1990.

[26] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1994.

[27] J. S. Kaufman, "Blocking in a shared resource environment," *IEEE Trans. Commun.*, vol. 29, pp. 1474–1481, Aug. 1981.

[28] E. Boros and A. Prékopa, "Closed form two-sided bounds for probabilities that at least $r$ and exactly $r$ out of $n$ events occur," *Math. Oper. Res.*, vol. 14, no. 2, pp. 317–342, 1989.

**Mingyan Liu** (S'96–M'00) received the B.Sc. degree in electrical engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1995, and the M.Sc. degree in systems engineering and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 1997 and 2000, respectively.

She joined the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, in September 2000, where she is currently an Assistant Professor. Her research interests are in performance modeling, analysis, energy-efficiency and resource allocation issues in wireless mobile ad hoc networks, wireless sensor networks, and terrestrial satellite hybrid networks.

Dr. Liu has been a member of the Association for Computing Machinery (ACM) since 2001.

**John S. Baras** (F'84) received the B.S. degree in electrical engineering from the National Technical University of Athens, Greece, in 1970, and the M.S. and Ph.D. degrees in applied mathematics from Harvard University, Cambridge, MA, in 1971 and 1973, respectively.

Since August 1973 he has been with the Electrical and Computer Engineering Department and the Applied Mathematics Faculty, University of Maryland, College Park, where he is currently a Professor and the Lockheed Martin Chair in Systems Engineering. He was the founding Director of the Institute for Systems Research (ISR, one of the first six NSF ERCs) from 1985 to 1991, and has been the Director of the Center for Hybrid and Satellite Communication Networks (a NASA Commercial Space Center) since 1991. His research interests include integration of logic programming and nonlinear programming for trade-off analysis, object oriented modeling, validation, and verification of complex systems models and engineering designs, hybrid, satellite and wireless communication networks, integrated network management systems, network security, stochastic systems, robust control of nonlinear systems, real-time parallel architectures for nonlinear signal processing, intelligent control systems, expert and symbolic systems for control and communication systems synthesis, biomimetic algorithms and systems for signal processing and sensor networks, intelligent manufacturing of smart materials. He has published over 350 technical articles on control, communication and computing systems, and was the editor of the book *Recent Advances in Stochastic Calculus* (New York: Springer, 1990).

Dr. Baras has served on the Board of Governors of the IEEE Control Systems Society, IEEE Engineering R&D Committee, Aerospace Industries Association advisory committee on advanced sensors, and IEEE Fellow evaluation committee. He is currently serving on the editorial boards of *Mathematics of Control, Signals, and Systems*, *Systems and Control: Foundations and Applications*, *IMA Journal of Mathematical Control and Information*, and *Systems Automation-Research and Applications*. He has received the 1978 NRL Research Publication Award, the 1980 Outstanding Paper Award of the IEEE Control Systems Society, the 1983 and 1993 Alan Berman Research Publication Awards from NRL, and the 2002 Best Paper Award at the 23rd Army Science Conference.