From: *Proceedings of the ACM SIGCOMM '94 Conference*; London, UK; August 31 to September 2, 1994.

Controlling Alternate Routing in General-Mesh Packet Flow Networks

Sandeep Sibal
ECSE Department
Rensselaer Polytechnic Institute
Troy, NY 12180-3590
sibal@networks.ecse.rpi.edu

Antonio DeSimone
Performance Analysis Department
AT&T Bell Laboratories
Holmdel, NJ 07733-3030
tds@hoserve.att.com

Abstract

High-speed packet networks will begin to support services that need Quality-of-Service (QoS) guarantees. Guaranteeing QoS typically translates to reserving resources for the duration of a call. We propose a state-dependent routing scheme that builds on any base state-independent routing scheme, by routing flows which are blocked on their primary paths (as selected by the state-independent scheme) onto alternate paths in a manner that is guaranteed—under certain Poisson assumptions—to improve on the performance of the base state-independent scheme. Our scheme only requires each node to have state information of those links that are incident on it. Such a scheme is of value when either the base state-independent scheme is already in place and a complete overhaul of the routing algorithm is undesirable, or when the state (reserved flows) of a link changes fast enough that the timely update of state information is infeasible to all possible call-originators. The performance improvements due to our controlled alternate routing scheme are borne out from simulations conducted on a fully-connected 4-node network, as well as on a sparsely-connected 12-node network modeled on the NSFNet T3 Backbone.

1 Introduction

Recent work on supporting real-time QoS applications in a packet switched environment suggests that reserving resources, particularly bandwidth, on a link is imperative [6]. Such a resource guarantee can be made possible by suitable service schemes such as Packetized Generalized Processor Sharing [35]. Also see [4, 13, 11, 39, 41] for other service schemes, and [40, 24] for a comparative survey. Packet networks supporting QoS move away from the pure datagram paradigm and best-effort service, to support flows that require some QoS guarantees with respect to transmission rate, delay, loss, etc. See [36] for example. A common factor in making QoS guarantees is the reserving of bandwidth. Resource reservations to support such QoS guarantees imply admission control: flows must be blocked from using a link whenever the inclusion of the flow will cause the total of the resources reserved to exceed the capacity of the link. Such a packet-switched network, which reserves resources for a flow along each link on a path, begins to look very much like a multi-rate circuit-switched network from the perspective of admission control and blocking on a link with respect to bandwidth.

Our motivation is the Internet of the not too distant future. The Internet today provides only best-effort service: admission control to guarantee a level of service does not exist, and network routing does not adapt to congestion. In the early days of the Internet, traffic-adaptive routing was an active area of experimentation. The early ARPA network routing algorithms used a traffic-sensitive, delay-based metric for routing—first using instantaneous delay measurements and a Bellman-Ford distance-vector algorithm[27] and then using averaged delay measurements and a Dijkstra shortest-path first algorithm[26]. The problems with this approach under heavy load became apparent, even for single-path routing[22], and the ARPANET began its move to a "capacity-based" metric (in the language of [22]). Modern routing protocols such as OSPF[31] emphasize quick adaptation to topology changes and low overhead, rather than adaptation to traffic, so that, in the Internet today, routes between an origin-destination pair are chosen based on traffic-insensitive metrics, which produce state-independent (SI) and, usually, single-path routes for all flows. The exceptions to single-path routing are routing using the IP TOS bits, and routing over equal-cost paths. Both exceptions are still state independent routings.

Soon, the Internet will begin to support real-time QoS calls, which will introduce flows to the network that have significant resource demands and require service guarantees. Current approaches to resource reservation[42] decouple the problem of resource reservation and admission control from the routing problem. We make the case that routing can be made to adapt simply and efficiently to congestion by applying the well-studied idea of *alternate routing* from telephony, to support flows blocked on their primary paths. Further, the importance of alternate routing strategies in improving network performance under extreme conditions is well-established by the experience in the AT&T network. Ash, *et al.* [2] present measurements of average network blocking performance in the AT&T switched network under extraordinary traffic conditions. For example, the Thanksgiving-day average network blocking dropped from 34% in 1986 to 3% in 1987 and to 0.4% in 1991, as more and more sophisticated routing strategies were implemented.

Still, for all the potential gains, experience with the operation of the the US telephone network going back at least to 1961 has demonstrated that networks with sophisticated alternate routing have complex and often undesirable behavior under overload[12]. We do not attempt to survey the vast telephony literature here. Kelly reviews the area in [19], and the interested reader is directed to this work and references therein for a comprehensive exposition on the underlying conceptual ideas and analytic results.

The value of alternate routing over state-independent routing is intuitively obvious. When there is insufficient bandwidth on a call's primary path, allowing a call to complete on an alternate path prevents the call from being lost. Alternate routing may be thus thought of as a scheme that exploits idle resources elsewhere in the network, caused by statistical fluctuations or imbalances of link loads. Because alternate routing *shares* resources more freely, it also has some desirable fairness properties which we will more clearly evidence in Section 4.

Less obvious are the problems that uncontrolled alternate routing can cause in a network. Careful study of alternate routing even under symmetric scenarios shows that uncontrolled alternate routing can actually do much worse than state-independent routing when the load on the network is beyond a certain critical (normalized) load. The exact value of this critical load depends on a host of factors: alternate path lengths, network size, graph structure of the network, etc. See [10, 19, 1, 25] for simulation and analytic studies of this phenomenon. The key to understanding why alternate routing can have a deleterious effect on the network is this: typically, an alternate-routed call will use more resources than a call routed on the primary path, and acceptance of an alternate-routed call can cause other calls to be blocked on their primary path and force them to a less efficient alternate path. This in turn can aggravate the problem of finding primary paths for other calls, resulting in an even larger fraction of calls choosing alternate routes, leading to even more inefficient utilization of resources. Such an avalanche effect drives the network into a high-blocking operating region. This behavior is not significant when the network is lightly loaded, since the probability of blocking on the primary path is small enough that the fraction of calls using alternate routes remains small, but under heavy loads the network can be driven into an inefficient state [1, 10].

Since uncontrolled alternate routing can lead to undesirable behavior at high loads, we need to find a mechanism that will tame such behavior. We choose to use *state-protection* as the technique to control such behavior. State-protection, also known as *trunk reservation*,[1] is a well-tested mechanism which has several desirable properties. Essentially it is a scheme that blocks alternate routed calls on a link, when its utilization is above a certain threshold. It can also be thought of as a method by which primary traffic is given *priority* over alternate traffic. Experiments show that the scheme is robust, in that a state-protection level optimized for a specific loading situation works well under variations in load. Key, in Section 2.2 of [21] demonstrates this property through an example. Robustness is important because the loading of links is *estimated* by the nodes they are incident on. For a discussion of the optimality of state-protection in certain environments, see [33].

In most theoretical studies in the field of circuit-switched routing, and in the important special case of the AT&T domestic long-distance network[2], the network is logically fully connected, and the primary path is the *direct* one-hop path. The alternate-routing problem is to select the best two-link path when the first choice is not available. Algorithms such as LBA and ALBA [28, 29], Dynamic Alternate Routing [9], and FAR [30] are notable approaches to the problem. See the work of Kelly [19] and Hunt & Laws [16] for some interesting asymptotic results. We are however interested in general-mesh networks, and these techniques do not allow for such extensions if global state information is not permitted, or if the primary path choice is fixed a priori according to some arbitrary (state-independent) routing rule. The environment we envision for our high-speed network is one where either: (1) the network has several links that are geographically distributed so that timely access to such global state information is impractical – unlike a fully-connected network where all links that comprise potential paths are typically at most a hop away, or (2) where it is desirable to build on an

[1] We hesitate to introduce new jargon but the term *trunk reservation* is problematic here for a couple of reasons. The idea of reservation here is distinct from reservation of resources for a call, and we do not want to overload the term. We also envision a more general context of a transmission medium than that indicated by the rather archaic term *trunk*.

already existing state-independent scheme.

Our work closely resembles the work of Ott and Krishnan [34], and it is instructive to compare their work with ours at this point. Their approach centers on the notion of a *shadow price*, which is the increase in calls lost on average in the future due to the acceptance of a call on a specific path when the network is in a given state. In their work, they approximate the shadow price for a call by summing over a simple shadow price associated with each link on the path. The approximation is the result of a certain *separability assumption*. The routing rule is then to accept a call along the path that minimizes this shadow price, unless this minimum is larger than the revenue that the call brings, in which case the call is blocked. The calculation of the shadow prices itself depends on the base routing policy used, but the *policy improvement lemma*[15] guarantees that the routing rule that results from their algorithm must improve on the base routing policy.

Our work differs in three conceptual aspects. First, we calculate the shadow price of accepting a call with the alternate routing scheme already in place, as opposed to their work where the shadow price is computed with respect to the base policy (following which the policy improvement lemma is invoked). Second we search for an upper bound on, instead of an approximation to, the shadow price of accepting a call on a specific route. Third, in our scheme, a route is chosen according to the base routing policy unless the base routing policy suggests a path that is blocking. It is only under such a scenario that the alternate-routing component kicks in. The state protection levels are set such that alternate routing is allowed only when the the revenue that the call (which would be blocked under the base policy) brings, is greater than the bound on the shadow price of accepting the call on the alternate path in question.

The separability assumption does not appear to jeopardize matters in [34], probably because their work focuses on the fully-connected telephony scenario, where the one-hop path between every origin-destination (O-D) pair is overwhelmingly chosen. In a general mesh setting this assumption appears tenuous, as is evidenced in Section 4 with respect to the NSFNet T3 Backbone.

Other pieces of work that are relevant are those of Dziong and Mason [7] and Kelly [20]. Dziong and Mason use an approach very similar to [34], but they employ the policy improvement lemma repeatedly to yield successively better policies in a continual manner. Kelly's work [20] on state-independent routing, can be extended to approximate the shadow prices for an alternate-routing scenario as well, though it is assumed therein that alternate-routed traffic is state-independent – an assumption that is questionable as Zachary notes in the introduction of [38].

To recapitulate, our routing approach involves the use of a base state-independent (SI) scheme as a first tier, augmented with a completely localized second state-dependent (SD) tier, which is applicable to general-mesh networks. In this preliminary study we do not address the support of multiple call types or multicast calls.

The scheme we propose works in the following fashion. A call request is made by a origin or the destination. The request specifies the origin, destination and a flow-rate. A *call set-up* packet containing the origin and destination node addresses, the flow-rate desired, and a *primary call* flag which is set, zips along the primary path[2] checking to see whether sufficient resources exist on each link of the primary path. If they do, resources are booked on its way back, and the call commences. If resources are not available on the primary path, alternate paths are successively attempted by *call set-ups* (whose primary path flags are reset) in order of increasing length.

In this study we demonstrate our control scheme with the minimum-hop SI routing rule. We have also studied a SI policy which minimizes link loss. The results are discussed in Section 4, but omit a detailed discussion due to space constraints. The minimum-hop SI routing policy is not usually an optimal SI routing policy, but in the context of alternate routing it appears to perform well. The choice of such an ad-hoc routing rule is partly deliberate – the idea being to show how alternate routing controlled in the way we describe can utilize idle resources in an excellent way. Further, it is well known that minimum-hop paths can be themselves computed in a distributed fashion with ease, and are therefore attractive in their own right. The mechanism by which traffic is routed on alternate paths–through source routes, using per-connection state in the network, or by some hybrid mechanism—is largely independent of the control algorithms we develop here. We note that the Internet itself may move to a source-routed approach driven in part by the need for policy-based routing [8]. Attempting alternate paths in order of increasing length is again attractive because distributed computation of alternate paths based on hop-count can be deduced with surprising ease from distributed minimum-hop path information. This observation is due to Harshavardhana, Dravida and Bondi [14], who describe a distributed algorithm (DALFAR) that computes alternate routes. The decision to admit a call on an alternate path is distributed, and is based on the state of each link constituting the path. A link will accept the call, provided its state (level of utilization) is below a certain threshold. This threshold is computed by the link itself, and is based on its current estimate

[2] We expect that this kind of signaling traffic is given priority, and adequate resources are reserved for the flow of such *call set-ups*. The amount of bandwidth required for this purpose should be typically negligible.

of the resource demand on the link due to calls whose primary path traverses that link. The estimate can be found from the primary call set-ups that fly past the link, or from measurements of established calls. The estimation procedure is not detailed in this report.

In what follows we use the terms *uncontrolled alternate-routing* to denote the scheme where if primary routes are blocking, alternate routes are attempted in order, as long as there is bandwidth available on the alternate path; *controlled alternate routing* to denote our scheme where alternate routed calls complete only if some local conditions are met; and finally *single-path routing* where calls are permitted to complete on their primary paths alone – that is alternate routing is prohibited. Note that the term *single-path* in this context is used in a loose fashion. It does not imply that a specific call type between a specific O-D pair is always routed along a fixed path, but that the chosen route (picked independent of state, with some probability, between a suite of choices) is the only one that is tried.

Our routing scheme is based on an analytic result proved in Section 2. The call arrivals are assumed to be Poisson, but with state-dependent rates. The result states that no matter what intensity of alternate-routed calls arrive at a link, state-protection guarantees an upper bound on the overall lost primary-routed calls at the link due to the acceptance of an alternate-routed call. In Section 3 we use this result to develop a scheme to control alternate routing based on state-protection thresholds, and show how these thresholds are chosen. The choice is made in such a manner that we are guaranteed, under the Poisson assumptions, that we will always perform better than single-path routing in terms of the total number of calls accepted. At low loads the algorithm behaves like the uncontrolled alternate routing case, while at high loads the algorithm behaves like single-path routing. The experimental results in Section 4 demonstrate this with examples.

2 The Main result

We consider point-to-point calls, specified by their origin, destination and the bandwidth they demand. In this preliminary study we assume calls of identical statistics: exponential holding times of equal mean length (we scale time so that the mean value is unity), and demanding an equal amount of bandwidth.

The capacity of a link may thus be represented in terms of the number of calls it can simultaneously support. The state of a link is denoted by the number of calls currently in progress on the link.

The traffic matrix is a square matrix of size N (the number of nodes in the network), and is denoted by $\mathcal{T}$. $\mathcal{T}(i,j)$ is then the traffic demand in Erlangs of calls originating at node i, and destined for node j. Every ordered node pair (i,j), has a unique primary path which we denote by $P^*(i,j)$.

Consider a link k. Let Λ^k be the primary traffic demand on link k. So that:

$$\Lambda^k = \sum_{P^*(i,j):k\in P^*(i,j)} \mathcal{T}(i,j) \qquad (1)$$

Let C^k denote the capacity of link k and let r^k denote the state-protection (reservation) level of link k. That is, in the last r^k+1 states, namely $(C^k-r^k, C^k-r^k+1, \ldots C^k)$, the link k will not accept alternate-routed calls. Let L^k denote the increase in the number of primary calls lost, due to the acceptance of an alternate routed call.

The theorem that follows is proven under the following Poisson assumptions:

A1 Alternate-routed calls arrive in a Poisson fashion at each link. The arrival rate of these calls is an arbitrary function of the link state.

A2 Primary calls also arrive at a link in a Poisson but state-independent fashion.[3]

We can relax the above assumptions by using a proof based on Markov decision theory, but that proof is more involved and somewhat less intuitive. It also requires a mild regularity condition on the base state-independent policy which would be satisfied by any reasonable base policy. The interested reader is directed to [37] for such a proof.

Theorem 1 *If a link k uses a state-protection level r^k, then under assumptions A1 and A2, L^k satisfies the inequality:*

$$L^k \leq \frac{B(\Lambda^k, C^k)}{B(\Lambda^k, C^k-r^k)} \qquad (2)$$

Proof:

We drop the superscript k for the duration of this proof.

We follow the convention that a call is lost at the link along its path where it is first blocked. Since some calls may be lost at other links, the effective arrival rate of primary calls on a link is no greater than the primary traffic demand on the link. Let the effective arrival rate of primary calls at a link be ν ($\nu \leq \Lambda$).

The birth-death process for the link thus has departure rates from a state to its neighbor on the right are given by the vector: $\underline{\lambda} = [\nu+\lambda_0^{(o)}, \nu+\lambda_1^{(o)}, \ldots \nu+\lambda_{C-r-1}^{(o)}, \nu, \nu \ldots]$, where $\lambda_s^{(o)}$ is the *overflow*, or arrival rate of alternate-routed calls when the link is in state s. The death rates, or departure rates from a state to its neighbor on the left are of course $[0,1,2,3\ldots C]$. We denote the blocking probability of the link by the *generalized* Erlang Blocking function: $B(\underline{\lambda}, C)$.

[3] This condition may be reduced to one where the primary call arrival rate is state-dependent, with the rates a non-increasing function of the link state.

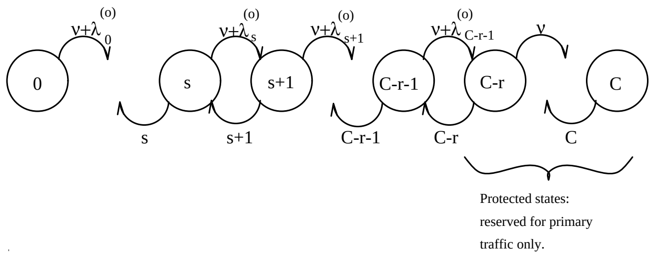

Figure 1: Markov Chain depicting the states of a link

Consider an alternate-routed call that arrives at time t, when the link is in state $s \in [0, C-r-1]$. If accepted, the link-state becomes $s+1$, if not it remains at s. Consider the case where the call is not accepted. Let the time at which the link-state will ultimately reach state $s+1$ for the first time (due to future primary or alternate-routed call accepts), be $t+\tau$. Then during the period $[t, t+\tau]$, no calls can be lost, and once in state $s+1$ (at $t+\tau$), the link behaves just like the case where the call was accepted at time t. If we push our horizon to infinity (assuming stationarity), the extra loss of primary calls when the call is accepted is:[4]

$$L = E[\tau] \cdot B(\underline{\lambda}, C) \cdot \nu \tag{3}$$

We now find a bound for $E[\tau]$. Let $X_{s,s+1}$, denote the expected number of accepted arrivals (primary or alternate-routed) from time t (when the link-state is s), until it first reaches state $s+1$.

$$X_{s,s+1} = \frac{\left(\nu + \lambda_s^{(o)}\right)}{s + \left(\nu + \lambda_s^{(o)}\right)} \cdot 1 + \frac{s}{s + \left(\nu + \lambda_s^{(o)}\right)} \cdot X_{s-1,s+1} \tag{4}$$

Since $X_{s-1,s+1} = X_{s-1,s} + X_{s,s+1}$, we have:

$$X_{s,s+1} = 1 + \frac{s}{\left(\nu + \lambda_s^{(o)}\right)} \cdot X_{s-1,s} \tag{5}$$

This recursive equation, along with the initial condition $X_{0,1} = 1$, yields the inverse Blocking function corresponding to a Markov Chain, $\mathcal{M}$, whose departure rates from a state to its neighbor on the right are given by the vector: $[\nu + \lambda_1^{(o)}, \ldots \nu + \lambda_{s+1}^{(o)}]$. Note that $\nu + \lambda_0^{(o)}$ is absent. The vector of death rates, or departure rates from a state to its neighbor on the left are: $[0, 1, 2, 3 \ldots s]$. Denote the associated Blocking function by $B_{\mathcal{M}}$, so that

$$X_{s,s+1} = [B_{\mathcal{M}}]^{-1} \tag{6}$$

Consider the Markov Chain $\mathcal{M}'$ derived from $\mathcal{M}$, where the death rates are increased by unity, so that the vector of death rates are now: $[1, 2, 3 \ldots (s+1)]$. Clearly:

$$B_{\mathcal{M}} \geq B_{\mathcal{M}'} \tag{7}$$

Now consider the Markov Chain $\mathcal{M}''$, derived from $\mathcal{M}'$, where an extra state is added *before* the states of $\mathcal{M}'$. Let the associated departure rate from it to its neighbor to its right be $\nu + \lambda_0^{(o)}$, and let it have a death rate of zero. Clearly then:

$$B_{\mathcal{M}'} \geq B_{\mathcal{M}''} \tag{8}$$

Note that $\mathcal{M}''$ looks exactly like a truncated version of the Markov Chain depicted in Figure 1. We thus use our earlier notation of the generalized Erlang Blocking function to denote $B_{\mathcal{M}''}$ by $B(\underline{\lambda}, s+1)$, where $\underline{\lambda} = [\nu + \lambda_0^{(o)}, \ldots \nu + \lambda_{s+1}^{(o)}]$. From Equations 6, 7 and 8, we thus have the result:

$$X_{s,s+1} \leq [B(\underline{\lambda}, s+1)]^{-1} \tag{9}$$

Since the inter-arrival time (in any state) is less than $1/\nu$, we have:

$$E[\tau] \leq [B(\underline{\lambda}, s+1) \cdot \nu]^{-1} \leq [B(\underline{\lambda}, C-r) \cdot \nu]^{-1} \tag{10}$$

From Equations 3 and 10, we thus have:

$$L \leq \frac{B(\underline{\lambda}, C)}{B(\underline{\lambda}, C-r)} \tag{11}$$

Denote $[B(\underline{\lambda}, x)]^{-1}$ by y_x, $x \in [C-r, C]$. We make use of the well-know recursion for the inverse Blocking function (see Equation 12 in [17]):

$$y_x = 1 + \frac{x}{\nu} \cdot y_{x-1} \tag{12}$$

Clearly then y_C is of the form:

$$y_C = f(\nu, r, C) + g(\nu, r, C) \cdot y_{C-r} \tag{13}$$

where $f(\nu, r, C)$ and $g(\nu, r, C)$ are positive valued. Note that $f(\nu, r, C)$ and $g(\nu, r, C)$ do not depend on the $\lambda^{(o)}$'s (overflow or alternate-routed call arrival rates). From Equation 13 it is clear that for fixed ν, r and C, y_C / y_{C-r} decreases with increasing y_{C-r}, that is $B(\underline{\lambda}, C)/B(\underline{\lambda}, C-r)$ decreases with increasing $B(\underline{\lambda}, C-r)$.

By pushing all $\lambda^{(o)}$'s to zero, we decrease $B(\underline{\lambda}, C-r)$, and thus increase $B(\underline{\lambda}, C)/B(\underline{\lambda}, C-r)$ for fixed ν, r and C. This implies:

$$L \leq \frac{B(\nu, C)}{B(\nu, C-r)} \leq \frac{B(\Lambda, C)}{B(\Lambda, C-r)} \tag{14}$$

The second inequality can be proven using arguments akin to those used earlier in the proof. QED.

3 The Routing Algorithm

A description of the overall routing mechanism has been outlined in Section 1. Here we focus on how we can use the result of Section 2 to come up with a smart

[4] This argument is the same as that of Ott and Krishnan[34]

state-protection level which seeks to optimize network performance, by minimizing the number of calls that are blocked, over a wide variety of loading patterns.

We are interested in enjoying the benefits of uncontrolled alternate routing at low to medium loads, without getting into high blocking states at high loads which are characteristic of uncontrolled alternate routing—see [1, 10, 25]. We will presently show that if the state-protection level (or reservation parameter) is chosen above a certain value we are *guaranteed* to do better than single-path routing, under the Poisson assumptions. This is particularly attractive, not only because we know we are necessarily improving on single-path routing, but because it is known that in most typical cases, single-path routing is near-optimal under suitably high loads. Our algorithm therefore uses the lowest reservation parameter (thereby imitating uncontrolled alternate routing to the utmost), that will guarantee that we always do better than single-path routing by the results of Section 2. We expect that for moderate loads, our algorithm will outperform uncontrolled alternate routing as well as single-path routing—a claim that is borne out in the results of Section 4.

3.1 Choosing the state-protection level

Consider a path P of an arbitrary alternate-routed call. Consider links $k \in P$. If we can guarantee that $\sum_{k \in P} L^k \leq 1$ for every P, we guarantee that by accepting the alternate routed call, we can only improve on the single-path routing policy. Denote the maximum hop-length over all alternate-routed calls by H. Note that as long as alternate paths are loop free, $H < N$, where N is the number of nodes in the network. Then clearly if $L^k \leq 1/H$ for all links k, this policy will always improve on the single-path routing policy. But from Theorem 1, we know that for a given link k, $L^k \leq B(\Lambda^k, C^k)/B(\Lambda^k, C^k - r^k)$. So as long as:

$$\frac{B(\Lambda^k, C^k)}{B(\Lambda^k, C^k - r^k)} \leq 1/H \text{ for all } k \qquad (15)$$

accepting an alternate call will necessarily improve on single-path routing. Note that for each link, C^k is known, Λ^k can be estimated, and H is a design parameter that is assumed fixed.[5]

From Equation 15, it is clear that if the inequality is satisfied for some value of r^k, it is satisfied for all larger values as well. We are interested in the smallest possible value of r^k that does not violate the inequality. Computations can be economized by using the recursive definition outlined in Equation 12.

We note that topology changes, and links going up or down, influence the computation of the state-protection level only insofar as it influences the primary traffic demand Λ on the link. Detection of such events, and their effects on the rest of the routing algorithm are outside the scope of this work.

Figure 2 shows values of r^k for $C^k = 100$, over the range $\Lambda^k \leq C^k$. The curves are drawn for $H = 2$, 6 and 120.

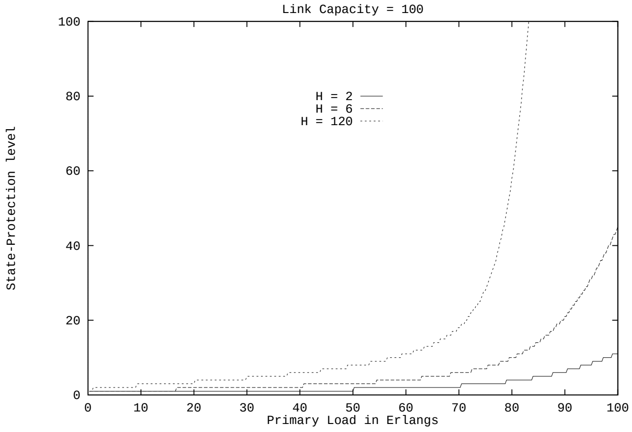

Figure 2: State-protection level r^k versus primary traffic load Λ^k, for a link k. Capacity $C^k = 100$, and maximum number of hops on alternate path routes $H = 2$, 6, 120

3.2 Discussion

Several observations are in order here. Note that H has nothing do with the length of primary paths. For the results of earlier sections to hold, a node pair could well have a primary path of length greater than H, though this would mean the absence of alternate paths for such a pair, since primary paths are of minimum hop length.

From Figure 2 it is clear that while r increases with H for a given load, the increase is *contained*. We have curves (not shown here) for $H \in [1000, 2000]$, for which $r \in [10, 20]$ for loads of 50 Erlangs (C=100).

It is worth comparing Mitra and Gibbens' result in [28] (see also [29]), where they study a fully-connected network with $H = 2$. When the primary (one-hop) path is blocked the least busy of the remaining two-hop alternates is chosen. We consider the case when C is 120 (the only one for which computed results appear in [28]). Their optimal r values for various number of alternates allowed, differ by at most two with respect to the results that we get at moderately high loads ($\Lambda \in [110, 120]$). This is the *crucial* range, because for loads less than this, the r values are sufficiently small that their influence on the dynamics of routing is minimal.

What if we chose $H < N - 1$? This would mean that fewer alternate routes will be available particularly for nodes further apart, implying an additional restriction on alternate routing. This is often not a serious concern

[5] H may be changed, but all links must be informed about it. It is also possible that each link k can pick its own H^k, which would be the maximum hop-length of alternate-routed calls that traverse link k. We do not study this possibility in this report.

even in a moderately sparse network as we shall see in Section 4, because there are typically so many alternate paths to begin with. The value of reducing H is that the r's can be pushed down, allowing freer alternate routing, which translates to better performance at low loads. An in-depth analytic study on how to choose a good value of H we leave as a topic for future research.

In our algorithm all alternate routed calls are treated equally — while shorter alternate paths are tried first, links will treat alternate calls of differing lengths the same. It is possible to prioritize shorter paths (as they are more resource efficient) by a state-protection scheme too, but this typically inflates the values of r for primary calls, and the gains tend to be overwhelmed by the losses due to the inflated r's, in the scenarios we have studied. We do not cover such schemes here.

Finally, it is worth noting that the strategy we have employed for controlling alternate routing, can be directly applied to other Multiple Service/Multiple Resource models as well, wherever alternate resource sets can be used (at an extra expense), when the primary resource or set of resources of a service is/are blocking. A good example is Channel Borrowing in Cellular Telephony [32, 18]. Here the resource is the channels in a cell instead of bandwidth in a link. The primary resource is the cell in which a call originates, and the alternate resource sets are the neighboring cells whose channels it can borrow. When a call arrives at a cell, which has no channels idle, a channel may be borrowed from a neighboring cell, but this will lead to locking of that channel in the *co-cells* of the borrowing cell. If a co-cell set consists of 3-cells (the situation most often discussed in the literature), then by choosing a r corresponding to $H = 3$, we can guarantee that Channel Borrowing will necessarily improve on the case when no Channel Borrowing is allowed. In a real scenario we expect such a scheme to be quite close to optimal, owing to the fact that the value of r for $H = 3$ will be quite small for $C \approx 50$.

4 Experiments

Call-by-call simulations were performed to test the performance of the control proposed in Section 3. Here we discuss two starkly different examples: a fully-connected symmetric 4-node network, and a sparsely connected 12-node network modeled on the NSFNet T3 Backbone. While the latter will be the focus, the former also illustrates the performance of the control proposed in Section 3.

The simulator, written in C, was run for 100 units of time. Recall that the call holding time is unity. It was run for each of 10 different seeds for a given traffic matrix $\mathcal{T}$. In addition each sample run was *warmed up* for 10 time units starting from an idle network. These simulation parameters were found to be sufficient for our examples. The algorithms studied were those of single-path routing, uncontrolled alternate routing and controlled alternate routing, according to the result of Section 3. Each algorithm was run with identical call arrivals and call holding times.

Note that we did not consider the case where links estimate Λ^k, an issue not covered in our work. We simply assumed that a link knew Λ^k a priori. This simplification should not take much from the validity of our results owing to the robustness of state-protection. See Section 2.2 in [21] for a discussion of why state-protection is a robust mechanism.

The Erlang Bound on the blocking probability was computed for both networks for each $\mathcal{T}$. The Erlang Bound is expected to be a rather loose lower bound, because it is a lower bound even if *re-packing* (rearranging existing calls) is allowed—something that would improve blocking performance but that we don't allow in any of our schemes. The Erlang Bound can be computed by evaluating the maximum of the following expression over all cut sets (S):

$$\frac{\sum_{\substack{i\in S\\ j\notin S}} \mathcal{T}(i,j)}{\sum_{i,j}\mathcal{T}(i,j)} \times B\left(\sum_{\substack{i\in S\\ j\notin S}} \mathcal{T}(i,j), \sum_{\substack{i\in S\\ j\notin S}} C(i,j)\right) +$$
$$\frac{\sum_{\substack{i\notin S\\ j\in S}} \mathcal{T}(i,j)}{\sum_{i,j}\mathcal{T}(i,j)} \times B\left(\sum_{\substack{i\notin S\\ j\in S}} \mathcal{T}(i,j), \sum_{\substack{i\notin S\\ j\in S}} C(i,j)\right)$$

Here $C(i,j)$ denotes the capacity of the link defined by the ordered pair (i,j), if the link exists. If i and j are not directly connected then $C(i,j) = 0$. For reasons of space we omit a discussion of the Erlang Bound and how it was efficiently computed in our scenarios. A proof of why this forms a lower bound on the overall blocking probability can be gleaned from Section 2.3 of [9] where the direction-less version is considered — that is, the node pair (i,j) does not have a notion of order.

4.1 Fully-connected Quadrangle

Figure 3 and Figure 4 show the blocking results for a fully-connected quadrangle, as a function of the offered load. The latter is a log plot to emphasize the blocking at low loads. The system with uncontrolled alternate routing performs well in the 85 Erlang and below range, and then the performance degrades badly. Single-path routing on the other hand does poorly up to 90 Erlangs, but then stays low. The controlled scheme however appears to stick with the better of the two, and performs better than either in the 85 to 95 Erlang range. Note that it validates our analysis in that our controlled alternate routing scheme does at least as well as single-path routing.

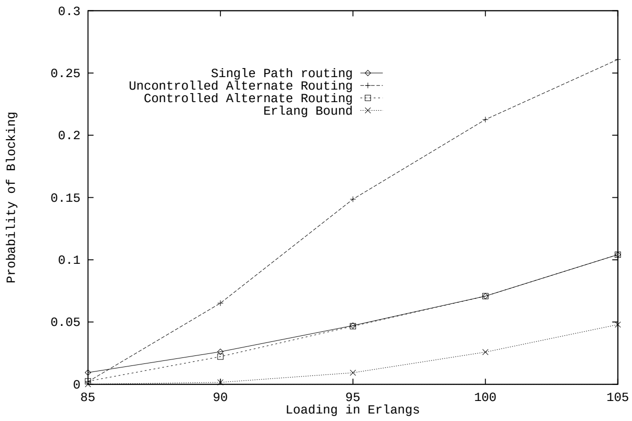

Figure 3: Blocking for a Fully-Connected Quadrangle

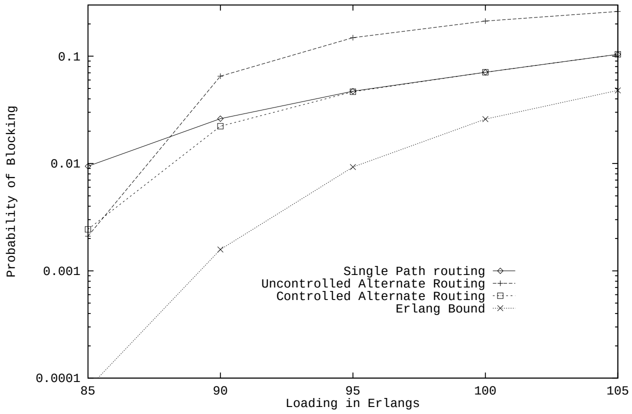

Figure 4: Blocking for a Fully-Connected Quadrangle

4.2 Internet

The NSFNet T3 Backbone was used as the model for this example. See Figure 5. The nodes here depict Core Nodal Switching Subsystems, and the names associated with each node (numbered from 0 to 11), correspond to the Exterior Nodal Switching Subsystems that connect to the Core Nodal Switching Subsystems. The map roughly corresponds to the configuration as of Fall 1992.

4.2.1 Setting up the network

We assumed that each link consists of a pair of unidirectional links transmitting in opposite directions. Forecasting into the future, we assumed a transmission rate of 155 Mb/s links each way, where 100 Mb/s has been allocated to rate-based traffic, and the remainder is consumed by best-effort traffic. A medium picture quality video call requiring 1 Mb/s was used as a prototype call. This means that $C = 100$ on each (directional) link.

Since the idea was to study a realistic scenario, the traffic matrix $\mathcal{T}$ given below, was computed to reflect the variety of load patterns on the current NSFNet T3 Backbone. This was done by using proportionality arguments (which we omit for reasons of space) starting from traffic estimates made in [5]. The elements of the matrix given below have been rounded to the nearest integer. This matrix was considered as the nominal load matrix. Note the wide disparities in the values of the elements of the traffic matrix, $\mathcal{T}$.

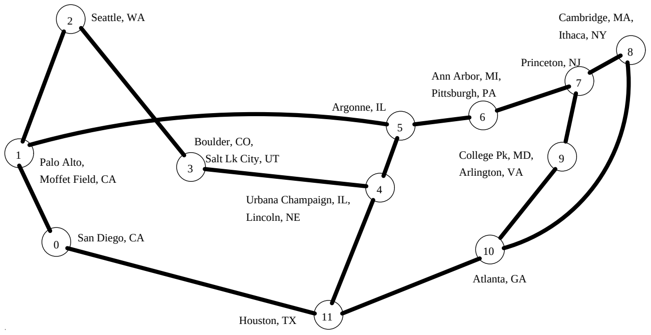

Figure 5: A map of the NSFNet T3 Backbone.

$$\begin{pmatrix}
0 & 4 & 2 & 3 & 6 & 0 & 7 & 1 & 9 & 5 & 2 & 3 \\
4 & 0 & 3 & 6 & 13 & 0 & 15 & 2 & 19 & 9 & 4 & 6 \\
2 & 3 & 0 & 2 & 5 & 0 & 6 & 1 & 7 & 3 & 1 & 2 \\
3 & 6 & 2 & 0 & 8 & 0 & 10 & 1 & 12 & 6 & 2 & 4 \\
7 & 14 & 5 & 9 & 0 & 0 & 24 & 3 & 31 & 15 & 6 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & 17 & 6 & 11 & 25 & 0 & 0 & 4 & 37 & 18 & 7 & 11 \\
1 & 2 & 1 & 1 & 3 & 0 & 3 & 0 & 4 & 2 & 1 & 1 \\
11 & 22 & 9 & 15 & 33 & 0 & 39 & 5 & 0 & 24 & 9 & 15 \\
5 & 9 & 4 & 6 & 14 & 0 & 16 & 2 & 21 & 0 & 4 & 6 \\
2 & 4 & 1 & 2 & 5 & 0 & 6 & 1 & 8 & 4 & 0 & 2 \\
3 & 6 & 2 & 4 & 8 & 0 & 10 & 1 & 12 & 6 & 2 & 0
\end{pmatrix}$$

Primary paths and (loop-free) alternate paths ordered by increasing length were computed using a K-*shortest path* algorithm. With the knowledge of the primary path and the above traffic matrix, the Λ^k's were calculated. See Equation 1. Two values of H were studied: $H = 11$ and $H = 6$. Note that the former allows arbitrarily long (loop-free) alternate paths, since $N = 12$. The values of r^k for both cases are tabulated in Table 1.

4.2.2 Simulation results

Figure 6 shows the blocking results for the 12-node network modeled after the NSFNet T3 Backbone. In Figure 6, a blocked call can attempt to complete on any non-looping path. On the average each node pair had about 9 alternate paths, with a maximum of 15 and a minimum of 5. The traffic matrix $\mathcal{T}$ was used for the nominal load, which corresponds to Load=10 in the

Link k	C^k	Λ^k	r^k	
			H=6	H=11
0→1	100	74	7	10
0→11	100	77	8	12
1→0	100	71	6	8
1→2	100	37	2	3
1→5	100	46	3	4
2→1	100	34	2	3
2→3	100	16	1	2
3→2	100	16	1	2
3→4	100	49	3	4
4→3	100	54	3	4
4→5	100	63	4	6
4→11	100	103	56	100
5→1	100	49	3	4
5→4	100	65	5	6
5→6	100	81	11	15
6→5	100	87	16	26
6→7	100	74	7	10
7→6	100	73	7	9
7→8	100	71	6	8
7→9	100	43	3	3
8→7	100	76	8	11
8→10	100	124	100	100
9→7	100	39	2	3
9→10	100	49	3	4
10→8	100	107	70	100
10→9	100	48	3	4
10→11	100	167	100	100
11→0	100	85	14	22
11→4	100	104	60	100
11→10	100	154	100	100

Table 1: Capacity (in Erlangs), primary load (in Erlangs), and state-protection levels for H=6 and H=11, of the (directed) links in the NSFNet T3 Backbone model, under the *nominal load* condition. Primary load values are rounded to the nearest integer.

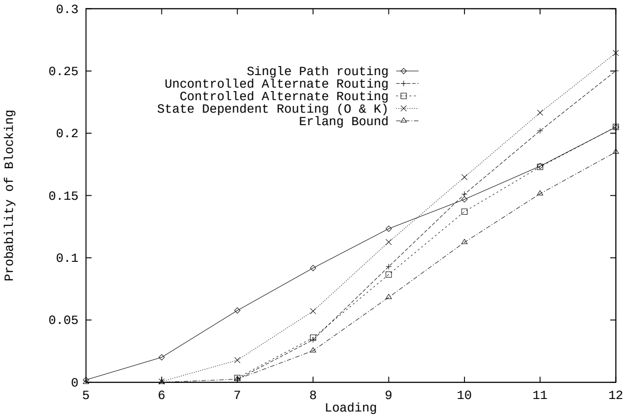

Figure 6: Internet Model: unlimited alternate path lengths

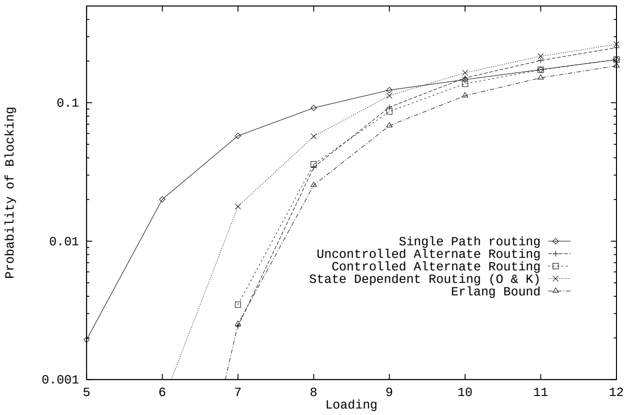

Figure 7: Internet Model: unlimited alternate path lengths

plot. The $\mathcal{T}$'s used for the other loads were got by linearly scaling the $\mathcal{T}$ corresponding to the nominal load. Figure 7 shows the same plot of the blocking probability, but on a log scale, to emphasize the behavior at low loads.

Figure 7 and Figure 6 show that single-path routing performs poorly compared to alternate routing at moderate loads, but approaches the Erlang Bound rapidly beyond that. Uncontrolled alternate routing, on the other hand performs very well and close to the bound for low loads, but poorly—worse than single-path routing—at loads above nominal. Our controlled alternate routing scheme does as expected: it improves performance under moderate loads, and, consistent with our analysis, never does worse than single-path routing. It is interesting to note that if the state-dependent scheme of Ott and Krishnan's [34] were to be used the performance is poor. We ascribe this to the fact that the approximate shadow prices that are computed using the separability assumption, swing more wildly when the network is sparse, so that choosing routes based on shadow price comparisons is more prone to error. In their work they use a reduced-load approximation to compute the *effective* primary load intensities. Here we have simply chosen to use the unreduced primary load intensities.

We have also investigated the effect of limiting the length of the alternate paths (the H parameter in Equation 15). When the alternate paths are limited to 6 hops, on the average each node pair had about 7 alternate paths with a maximum of 13, and minimum of 5. It is somewhat encouraging that, even by cutting down the maximum hop length by about half in a network that is reasonably sparse, the alternates available to a node-pair do not change in any drastic fashion. The 6-hop results show a small improvement of controlled alternate routing and little change in single-path and uncontrolled alternate routing. We attribute the improvement to a reduction in the values of r needed to satisfy Equation 15, while almost all the *good* alternate paths remain even when H is reduced to 6. This observation also tells us that the values of r computed in Section 3 (based on the result in Section 2) may be more conservative than they need to be.

We have carried out other simulation experiments to investigate the controlled alternate routing strategy.

Link failures We disabled links 2→3 and 3→2, and while the blocking in general was higher, the relative position of the curves was maintained. This was also the case when links 7→9 and 9→7 were disabled.

Blocking on an O-D pair basis Until now we have looked at average network blocking as a performance measure. The skewness in blocking probabilities across O-D pairs for the case $H = 6$ was also studied. As expected the blocking was most skewed for the single-path routing case, and least skewed for the uncontrolled alternate routing case almost uniformly across all O-D pairs. This reaffirms the observation made in Section 1 regarding the inherent fairness property of alternate routing, because of the greater degree to which it shares network resources.

Primary paths chosen to minimize link loss In all of the above we chose the minimum hop path as the primary path. We also re-ran all of the above experiments when primary paths were chosen so as to minimize overall system blocking of primary calls, under the independent link assumption. In general, this resulted in bifurcated primary flows, where a path would be a primary path for an O-D pair with a certain probability. The expected number of lost calls on a link of capacity C, fed by a Poisson stream of traffic intensity Λ, each call holding for unit time and requiring unit bandwidth, is convex in Λ. See [23] for a proof. Using this as a cost function we used an iterative conjugate-gradient method to minimize the expected sum of link costs [3]. The results for the case without alternate routing did better than in the minimum-hop primary path scenario. However when our alternate routing algorithm was added, the performance was almost coincident with that of the minimum-hop primary path scenario. This suggests that our scheme, at least in this example, is robust, in that it appears to be insensitive to the two differing ways of choosing primary paths.

5 Concluding remarks

This work demonstrates how the benefits of alternate routing – lower average network blocking, better fairness in blocking on a node pair basis, less sensitivity of blocking performance to traffic estimates and network engineering – can be exploited without moving the network into an inefficient state of operation by a simple distributed and robust control strategy. While state-protection (the control mechanism used) has been studied and algorithms have been suggested on how the state-protection levels are to be chosen for fully-connected networks, our work is applicable to a general-mesh network, and therefore applicable to candidate multi-path routing strategies that might be implemented on the Internet as it evolves to support traffic demanding resource reservations for guaranteed QoS. Our proposed state-dependent scheme can work in conjunction with any state-independent routing rule. The scheme is lightweight in that it does not require links to advertise state information, but simply requires knowledge by a node of the state of links incident on it. It also provides the important guarantee that the network will necessarily improve on the case when only state-independent routing is permitted, under the Poisson assumptions. The result in Section 2 is the pivotal argument that we use in designing our distributed routing algorithm. The power of the control scheme has been demonstrated in Section 4, both for a simple fully-connected network and for a more realistic network based on the NSFNet T3 Backbone topology and traffic.

References

[1] Akinpelu, J. M. The overload performance of engineered networks with non-hierarchial routing. *AT&T Bell Laboratories Technical Journal 63*, 7 (September 1984), 1261–1281.

[2] Ash, G. R., Chen, J.-S., Frey, A. E., Huang, B.-S. D., Lee, C.-K., and McDonald, G. L. Real-time network routing in the AT&T network—improved service quality at lower cost. In *Proceedings of IEEE Global Telecommunications Conference* (Orlando, FL, December 1992), pp. 802–813.

[3] Bertsekas, D., and Tsitsiklis, J. *Parallel and Distributed Computation.* Prentice Hall, 1989.

[4] Demers, A., Keshav, S., and Shenker, S. Analysis and simulation of a fair queueing algorithm. *Journal of Internetworking: Research and Experience 1* (1990), 3–26. Also in Proc. ACM SIGCOMM '89.

[5] DeSouza, O. Internet traffic projections. Work in progress.

[6] Doshi, B. T. Deterministic rule-based traffic descriptors for broadband ISDN: Worst-case behavior and connection admission control. In *Proceedings of IEEE GLOBECOM '93* (1993), pp. 1759–1764.

[7] Dziong, Z., and Mason, L. An analysis of optimal call admission and routing model for multiservice loss networks. In *Proceedings of IEEE INFOCOM '92* (1992), pp. 141–152.

[8] Estrin, D., Steenstrup, M., and Tsudik, G. A protocol for route establishment and packet forwarding across multidomain internets. *IEEE/ACM Trans. on Networking 1*, 1 (February 1993), 56–70.

[9] Gibbens, R., and Kelly, F. Dynamic routing in fully connected networks. *IMA Journal of Mathematical Control and Information 7* (1990), 77–111.

[10] Gibbens, R. J., Hunt, P. J., and Kelly, F. Bistability in communication networks. In *Disorder in physical systems*, G. Grimmett and D. Welsh, Eds. Oxford University Press, 1990, pp. 113–128.

[11] Golestani, S. J. A framing strategy for congestion management. *IEEE Journal on Selected Areas in Communications 9*, 7 (September 1991), 1064–1077.

[12] Haenschke, D. G., Kettler, D. A., and Oberer, E. Network management and congestion in the U.S. telecommunications network. *IEEE Transactions on Communications COM-29*, 4 (April 1981), 376–385.

[13] Hahne, E. H. Round-robin scheduling for max-min fairness in data networks. *IEEE Journal on Selected Areas in Communications 9*, 7 (September 1991), 1024–1039.

[14] Harshavardhana, P., Dravida, S., and Bondi, A. B. Congestion control for connectionless networks via alternate routing. In *Globecom '91* (Phoenix, Arizona, December 1991), IEEE Global Telecommunications Conference, IEEE, pp. 339–346.

[15] Howard, R. *Dynamic Programming and Markov processes.* Wiley, New York, 1960.

[16] Hunt, P., and Laws, C. Asymptotically optimal loss network control. *Mathematics of Operations Research 18*, 4 (1993), 880–900.

[17] Jagerman, D. L. Methods in traffic calculations. *AT&T Bell Laboratories Technical Journal 63*, 7 (September 1984), 1283–1310.

[18] Johri, P. K. A note on the dynamic channel assignment in cellular radio networks. Tech. Rep. 45312-91108-01 TM, AT&T, 1991.

[19] Kelly, F. Loss networks. *The Annals of Applied Probability 1* (1991), 319–378.

[20] Kelly, F. P. Routing in circuit-switched networks: optimization, shadow prices and decentralization. *Advances in Applied Probability 20* (1988), 112–144.

[21] Key, P. B. Optimal control and trunk reservation in loss networks. *Probability in the Engineering and Informational Sciences 4* (1990), 203–242.

[22] Khanna, A., and Zinky, J. The revised ARPA routing network. *ACM Computer Communications Review* (1989), 45–46.

[23] Krishnan, K. R. The convexity of loss rates in an Erlang loss system and sojourn in an Erlang delay system with respect to arrival and service rates. *IEEE Transactions on Communications 38*, 9 (September 1990), 1314–1316.

[24] Kurose, J. Open issues and challenges in providing quality of service guarantees in high-speed networks. *Computer Communication Review* (January 1993), 6–15.

[25] Mason, L. G. On the stability of circuit-switched networks with non-hierarchial routing. In *Proc. 25th Conference on Decision and Control* (Athens, Greece, December 1986), pp. 1345–1347.

[26] McQuillan, J. M., Richer, I., and Rosen, E. C. The new routing algorithm for the internet. *IEEE Trans. on Communications 28*, 5 (1980), 711–719.

[27] McQuillan, J. M., and Walden, D. C. The ARPA network design decisions. *Computer Networks* (1977), 243–289.

[28] Mitra, D., and Gibbens, R. J. State-dependent routing on symmetric loss networks with trunk reservations: Analysis, asymptotics, optimal design. Tech. Rep. 11212-900703-22 TM, AT&T, 1990.

[29] Mitra, D., Gibbens, R. J., and Huang, B.-S. D. Analysis and optimal design of aggregated-least-busy-alternative routing on a symmetric loss network with trunk reservations. In *Proceedings of the* 13$^{\text{th}}$ *International Teletraffic Congress* (1991), A. Jensen and V. B. Iversen, Eds., Elsevier North-Holland, pp. 477–482.

[30] Mitra, D., and Seery, J. B. Comparative evaluations of randomized and dynamic routing strategies for circuit-switched networks. *IEEE Trans. on Communications 39*, 1 (January 1991), 102–116.

[31] Moy, J. OSPF version 2. Tech. rep., July 1991. RFC 1247.

[32] Nanda, S., and Goodman, D. J. Dynamic resource allocation: A scheme for carrier allocation in cellular systems. Tech. Rep. 45312-910405-01 TM, AT&T, 1991.

[33] Nguyen, V. On the optimality of trunk reservation in overflow processes. *Probability in the Engineering and Informational Sciences 5* (1991), 369–390.

[34] Ott, T., and Krishnan, K. State dependent routing of telephone traffic and the use of separable routing schemes. In *Proc. 11th International Teletraffic Congress* (Kyoto, Japan, 1985).

[35] Parekh, A. K., and Gallager, R. G. A generalized processor sharing approach to flow control. Tech. Rep. 2040 & 2076, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 1991.

[36] Partridge, C. A proposed flow specification. Tech. rep., September 1992. RFC 1363.

[37] Sibal, S. A decomposition theorem for loss networks with applications to distributed routing. Unpublished work, 1994.

[38] Zachary, S. On blocking in loss networks. *Advances in Applied Probability 23* (1991), 355–372.

[39] Zhang, H., and Ferrari, D. Rate-controlled static-priority queueing. In *IEEE INFOCOM* (San Francisco, CA, April/May 1993), pp. 227–236.

[40] Zhang, H., and Keshav, S. Comparison of rate-based service disciplines. In *ACM SigComm* (Zurich, Switzerland, September 1991), pp. 113–122.

[41] Zhang, L. Virtual clock: A new traffic control algorithm for packet switching networks. In *ACM SigComm* (Philadelphia, PA, September 1990), pp. 19–29.

[42] Zhang, L., et al. A new resource reservation protocol. *IEEE/ACM Trans. on Networking 1* (September 1993).

Contents

List of Figures

List of Tables