

Naval Research Laboratory

Washington, DC 20375-5320

AD-A261 055



NRL/MR/5521--93-7196

Admission Control in Integrated Voice/Data Multihop Radio Networks

CRAIG M. BARNHART AND JEFFREY E. WIESELTHIER

*Communication Systems Branch
Information Technology Division*

ANTHONY EPHREMIDES

*Locus, Inc.
Alexandria, Virginia*

AND

*University of Maryland
College Park, Maryland*

January 18, 1993

**DTIC
ELECTE
MAR 09 1993
S E D**

93-04967



5408

88

88

099

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE January 18, 1993	3. REPORT TYPE AND DATES COVERED Interim Report 12/91-12/92		
4. TITLE AND SUBTITLE Admission Control in Integrated Voice/Data Multihop Radio Networks			5. FUNDING NUMBERS PE-61153N PR-RR015-09-41 WU-DN159-036	
6. AUTHOR(S) Craig M. Barnhart, Jeffrey E. Wieselthier and Anthony Ephremides				
7. PERFORMING ORGANIZATION NAME(S) and ADDRESS(ES) Naval Research Laboratory Washington, DC 20375-5320			8. PERFORMING ORGANIZATION REPORT NUMBER NRL/MR/5521-93-7196	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research Arlington, VA 22217			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES A. Ephremides is with Locus, Inc. and the University of Maryland.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>In this report, we investigate the admission control problem for voice traffic in integrated multihop radio networks. We consider only those admission control policies that yield a "coordinate convex" state space. This restriction, in conjunction with the use of a "blocked-calls-cleared" mode of operation and the assumption that the voice process is Markovian, results in a product-form stationary distribution for the voice state of the system. The product-form distribution allows the straightforward evaluation of network performance when subject to different admission control policies.</p> <p>A large number of coordinate convex admission-control policies must be searched to determine the optimal policy. We develop a recursive procedure to speed the evaluation of a large number of different policies, as well as descent-search method to reduce significantly the number of policies that must be evaluated in searching for the optimal one. The numerical examples we present indicate that improved performance can be obtained by administering admission control, but this improvement is typically small unless different revenues or costs are associated with the various call types.</p>				
14. SUBJECT TERMS Communications Network Circuit-Switched Network			15. NUMBER OF PAGES 56	
Admission Control			16. PRICE CODE	
Product-Form Solution				
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

CONTENTS

1	INTRODUCTION	1
1.1	Outline of the Report.....	5
2	A VOICE-ONLY MODEL.....	6
2.1	Problem Formulation.....	6
2.2	Control Policy	8
2.3	The Product-Form Distribution.....	12
3	COMPUTATIONAL ISSUES IN EVALUATING PRODUCT-FORM SOLUTIONS	14
3.1	A Recursive Procedure	14
3.2	Descent Search and the Question of Unimodality.....	16
3.2.1	Descent Search	18
4	NUMERICAL EVALUATION OF NETWORK PERFORMANCE	19
4.1	A 10-Node, 5-Circuit Example.....	20
	The Unimodality Issue Revisited	24
4.1.1	Blocking Cost Minimization: A Weighted Performance Index.....	26
4.1.2	Increased Capacity.....	27
4.1.3	Increased Congestion	28
4.2	Routing and Admission Control	30
5	AN EXAMINATION OF SPECIAL NETWORK CLASSES	35
5.1	Analysis of Tandem Networks.....	35
5.1.1	A Five-Node Tandem.....	36
	A Progressive-Search Method	39
5.1.2	A Four-Node Tandem	40
5.2	Multi-Cross Networks and Network Self Regulation.....	41
6	EXTENSION TO INTEGRATED NETWORKS.....	44
7	CONCLUSIONS	46
	REFERENCES.....	47
	APPENDIX	49

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

ADMISSION CONTROL IN INTEGRATED VOICE/DATA MULTIHOP RADIO NETWORKS

1 INTRODUCTION

Voice and data impose conflicting requirements on the networks that must support them. For example, voice communication requires that there be very low variability of the time delay, so that a continuous voice stream is presented to the user at the destination; typically, delivery in nearly real time is required.¹ In contrast, data traffic can often tolerate both long delay (except for certain time-critical messages) and high variability of delay. Also, data transmission requires very low probability of error to ensure the preservation of data integrity, whereas voice communication can tolerate considerably higher error rates as a result of the inherent redundancy of speech signals.

Traditionally, separate networks have been used to handle voice and data traffic. However, greater efficiency can be achieved if the available network resources are shared intelligently by these two classes of traffic. The voice/data integration problem is an example of a much broader class of common engineering problems that deal with efficient use of a limited set of resources to meet certain requirements of service. In this problem the resource set is the communication channel (i.e., bandwidth, which may include a mixture of diverse communication media such as HF and UHF terrestrial links, SATCOM, etc.) and the hardware associated with maintaining and using the channel (e.g., transmitters, receivers, buffers); the demands are voice traffic and data traffic. The use of common resources for these different classes of traffic has the potential not only to reduce the resources needed to support user demands, but also to provide improved performance through the adaptive sharing of resources.

The different requirements for the voice and the data processes motivate the customary approach of establishing a circuit-switched path (i.e., either a true circuit or a virtual circuit) between communicating nodes for the duration of a voice call,² and using packet switching with queueing (at source and relay nodes) for data traffic. Because the source and the destination nodes are not generally within direct communication range of each other, the use of relaying over multihop paths is required for both classes of traffic. We

¹ Although it would certainly be possible to consider systems in which voice messages are stored for forwarding at a later time, either at the source node or at an intermediate node, we do not address this possibility in this report. Typically, delays greater than 0.25 s are considered unacceptable.

² In this report, the terminology of "call" always refers to a voice call, whether or not voice is explicitly mentioned.

assume that voice calls that are not accepted for immediate transmission (in practical terms, this may mean within several hundred ms of their arrival) are "blocked," i.e., lost from the system, a mode of operation generally referred to as "blocked calls cleared."³ Appropriate performance measures for this mode of operation include blocking probability and throughput. Performance measures appropriate for data traffic include delay and throughput. A common performance criterion used in integrated networks combines voice and data measures to yield the weighted sum of voice-call blocking probability and data-packet delay.

Network performance can be improved by administering controls in the form of call admission policies for voice traffic, as well as by routing and by link-activation scheduling [1, 2] for both voice and data. These are complex, interdependent problems that must be addressed to determine the form of the "optimal" integration scheme. Each is a complex problem in its own right that has been (for the most part, separately) examined in homogeneous network settings (i.e., voice-only or data-only networks) [3]. In this report we recognize the interdependence of these problems, but focus primarily on the voice-call admission control problem. We eliminate the need to solve the routing problem by assuming the use of fixed multihop paths between source-destination pairs, although we do touch upon the issue of choosing good paths. Similarly, we eliminate the need to address the link-activation scheduling problem by assuming the use of channelization in the frequency domain (by means of frequency-division multiple access—FDMA) rather than in the time domain (by means of time-division multiple access—TDMA).⁴

Although a great deal of attention has been paid to the modeling and performance evaluation of circuit-switched voice in communication networks [4, 5], relatively little has been done until recently on the question of voice-call admission control in such systems. An uncontrolled mode of operation is typically assumed, in which all voice calls are admitted as long as sufficient network resources (e.g., link bandwidth) are available; calls are blocked if, and only if, resources are not available for them. Accepting a call implies a long-term commitment of the resources in the circuit (i.e., until the call is completed); it effectively reduces the resources available to other voice calls and to data. In a controlled system, it may be advantageous to block a call even though resources are currently available because the acceptance of a particular call now may result in the blockage of several other future calls (or perhaps a call of higher precedence) that could otherwise have been accepted, and may cause long queueing delays for data traffic. A clever network administrator may be able to

³ Although not necessarily the best from a practical standpoint, this mode of operation results in a mathematically tractable system model that is amenable to numerical evaluation.

⁴ Later in this section we discuss briefly the differences between FDMA and TDMA operation.

improve performance by selectively blocking some calls even though resources are available, or by reserving a certain portion of the resources for data only.

In [6] Viniotis and Ephremides show that, for a single node, the optimal access control takes the form of a switching curve, i.e., the decision on whether to accept a voice call depends monotonically on the number of voice calls already in progress and on the data-packet queue size. The detailed structure of the switching curve depends on factors such as traffic statistics and the cost function (e.g., the relative weighting of voice-call blocking probability and data-packet delay). They extended their results to certain tandem network configurations [7], but no results of this type have been obtained for more general network configurations. An example of a suboptimal access-control scheme is the "boundary" approach [8], under which the network's resources are divided into two compartments, one for voice traffic and the other for data traffic. Voice calls are admitted as long as the channels (e.g., frequency subbands or time slots) in the voice compartment are not all occupied. The position of the boundary, which is independent of the data-packet queue size, is chosen to minimize the specified performance criterion. Under the "movable-boundary" version of this scheme, data traffic is permitted to use any idle portion of the voice compartment with the understanding that the subchannel will be returned to voice upon demand. This is possible because the transmission of a data packet requires only a short-term commitment of resources. Voice traffic is independent of data traffic in such systems (since the position of the boundary does not depend on queue size), although data performance depends on the voice traffic because the resources available for data depend on those being used by voice.

The problem of call admission is considerably more complicated in multihop networks. In this report we focus our attention on the particular case of circuit-switched multihop wireless networks, for which the question of admission control has been virtually unaddressed; in fact, there have been only few attempts to develop circuit-switched models that incorporate the properties of wireless networks, and these have been for cellular radio and single-hop satellite communication rather than general multihop configurations. In particular, we endeavor to show how the recently introduced methodology of multiple-service, multiple-resource (MSMR) modeling, in conjunction with the use of "coordinate convex" policies [9, 10], can be used to study voice admission control in radio environments. These policies base admission-control decisions on the state of the network, i.e., on the set of currently active voice calls (and not only on whether or not resources are available to support the call). The potential advantage of state-dependent admission control is the possibility of performance gains in terms of overall blocking probability (or, equivalently, throughput) if calls can be selectively rejected even in the presence of available resources.

Although coordinate-convex policies are not, in general, optimal, our studies demonstrate that they are quite effective. Under reasonable modeling assumptions, the MSMR methodology leads to a product-form probability distribution for system state, which permits system evaluation without solving the balance equations of the underlying Markov chain. However, the limitations of this approach become evident through the explosive growth of the computational complexity as the network size increases. In this report we propose efficient, suboptimal techniques that stem from the MSMR formulation and that can be applied to wireless networks.

The key property that distinguishes the multihop radio environment from wireline networks is the broadcast nature of radio transmission, which permits each transmission to be heard by all of the transmitting node's neighbors. A negative consequence of the broadcast property is the possibility of interference, which necessitates the scheduling of a node's transceiver resources in coordination with those of its neighbors; in this study we assume the use of orthogonal FDMA channelization, which simplifies the scheduling issue greatly. We assume that enough frequency subchannels are available so that each node has several transmission frequencies for its exclusive use;⁵ a node tunes one of its receivers to the appropriate transmitter frequency of each of the neighbors it wants to listen to. This problem can also be addressed in the time domain, in which case it is generally known as link-activation scheduling [1, 2]. Typically, a single frequency channel is divided on a time-division basis, with frequency reuse exploited by nodes sufficiently distant from each other. This problem is considerably more difficult than the frequency-domain version just discussed, and most versions of it are NP-complete (i.e., cannot be solved in polynomial time). The increased difficulty arises from the fact that each node must coordinate its link-activation schedule (i.e., the time slots in which its transmissions and receptions take place) with those of all of its neighbors.⁶

On the other hand, a beneficial feature that arises from the broadcast mode of transmission is the flexibility to support a time-varying set of communication links in response to changing topology and changing communication requirements; a link can be formed between two nodes as long as they are within communication range of each other,

⁵ This assumption may be relaxed to permit frequency reuse in distant parts of the network.

⁶ The direct dual of this problem in the frequency domain would be the case in which only a small number of frequencies are available (instead of a set of unique frequencies at each node), in which case the difficulty of the frequency allocation problem would be comparable to that of link activation. Our assumed availability of a sufficient number of frequencies eliminates the need to coordinate frequency use, and thus eliminates the scheduling problem.

provided that transceivers are available to do so. These properties are discussed in greater detail in Section 2.

In addition, multihop radio networks differ vastly from wireline telephone networks, both in size (number of nodes) and in the number of channels available at each node. Several approximation techniques that work well for the case of trunk-type links are based on the capability of trunk lines to support a large number of calls simultaneously. However, Geraniotis has recently shown [11] that the stochastic knapsack and Pascal approximations studied by Ross et al. [12, 13] provide fairly good agreement for smaller systems also, thus suggesting that they may be applicable to multihop radio networks as well.

Our study is actually intended as a stepping stone toward modeling integrated voice-data service on multihop radio networks. The idea of integration in wireless networks that we have proposed [14] relies on the use of "boundary" techniques that allocate a portion of the channel bandwidth to the voice traffic and the remainder to data. The resulting voice "sub-network" can be analyzed in the manner described here. Thus our approach can be subsequently augmented to address the data portion of the network. We discuss briefly how the residual capacity that is unused by voice traffic can be used to support data traffic.

1.1 Outline of the Report

In Section 2 we develop a model for multihop voice communication in radio networks, and formulate the problem of optimal voice admission control in such a network. Under certain reasonable assumptions, which are discussed, a "product-form" solution can be applied to this model to measure network performance characteristics such as blocking probability and throughput.

In Section 3 we discuss some of the computational issues associated with evaluating product-form solutions. A recursive procedure is developed to accelerate the evaluation of a large number of different admission control policies, and a descent-search method is developed to minimize the number of policies that must be evaluated in searching for the optimal one. In Section 4 we present numerical examples of the effects of different control policies on network performance measures such as blocking probability and throughput. The benefits of admission control policies and route selection algorithms are also compared.

In Section 5 we discuss two important classes of networks that are either building blocks of multihop networks or display significant behavior. First we consider tandem networks, and demonstrate the high degree of complexity that arises even in relatively simple networks. We also consider "multi-cross networks," in which several mutually disjoint circuits intersect (and therefore compete for resources) with a multihop circuit.

In Section 6 we introduce the concept of "residual data capacity," and discuss how this capacity that is unused by voice traffic can be used to support data traffic. We show how our circuit-switched voice models can be used to obtain a measure of the residual data capacity, and consequently, a metric that allows simultaneous optimization of both the data and voice portions of an integrated network.

Our conclusions, which are summarized in Section 7, indicate that the application of admission control can, in fact, lead to improved performance. However, this improvement is of significant proportions only if the various call types are not all weighted equally. If they are equally weighted, any attempt to perform sophisticated admission control adds very little improvement to simple admission policies that accept all calls as long as there are available network resources. We also address the impact of the choice of path sets used to implement the circuits, and we show that the use of good path sets can be more important than the admission control policy.

2 A VOICE-ONLY MODEL

We first consider a radio network with only voice traffic, in which a multihop circuit is established between the source and destination nodes throughout the duration of the call. Data traffic can then be integrated into this model as discussed in Section 6.

2.1 Problem Formulation

Network topology can be described in terms of the communication resources available at each node and the connectivities between nodes, where FDMA is used to provide contention-free channel access. To illustrate our problem formulation, we consider the simple seven-node star-network example shown in Fig. 1. Nodes 2 - 7 are each connected to only node 1, thus necessitating the use of node 1 as a relay in all multihop circuits. We consider three particular source-destination pairs (2-5, 3-6, and 4-7), which correspond to Circuits 1, 2, and 3, respectively. The state of the system is defined to be $x = \{x_1, x_2, x_3\}$, where x_i is the number of calls currently active over Circuit i .

In an uncontrolled system, a voice call is accepted as long as there are sufficient resources at all nodes along the multihop path. For example, if node i has T_i transceivers, it can support up to T_i simultaneous calls.⁷ We refer to the limits imposed by the values of T_i

⁷ Other models are certainly possible. For example, if traffic is one way, rather than interactive, the source node would not have to dedicate a receiver to support the call. Similarly, receive-only nodes would not need transmitters. The approach presented here can be modified straightforwardly to accommodate variations such as these. The implicit assumption that the number of transmitters and receivers at a node are equal, which lets us describe the node in terms of a single parameter T_i , is used here because of convenience of notation, and can easily be relaxed.

as the "system constraints." For example, if each of the nodes shown in the three-circuit network of Fig. 1 has five transceivers, the resulting uncontrolled system state space Ω is as shown in Fig. 2. Equivalently, the system constraints expressed as inequalities describe the system state space:

$$\begin{aligned} x_1 \leq 5; \quad x_2 \leq 5; \quad x_3 \leq 5; & \quad (\text{i.e., no more than 5 calls may be accepted on any circuit}) \\ x_1 + x_2 + x_3 \leq 5. & \quad (\text{the hub, node 1, can handle no more than 5 calls}) \end{aligned}$$

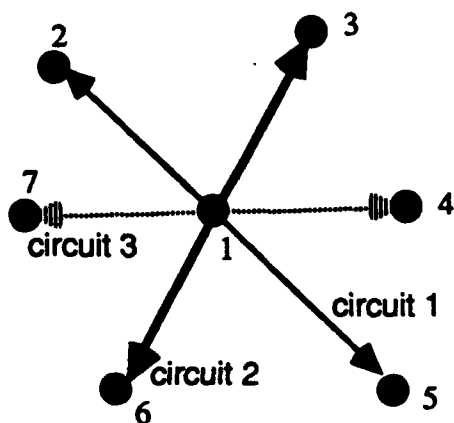


Fig. 1 — A simple 3-circuit network

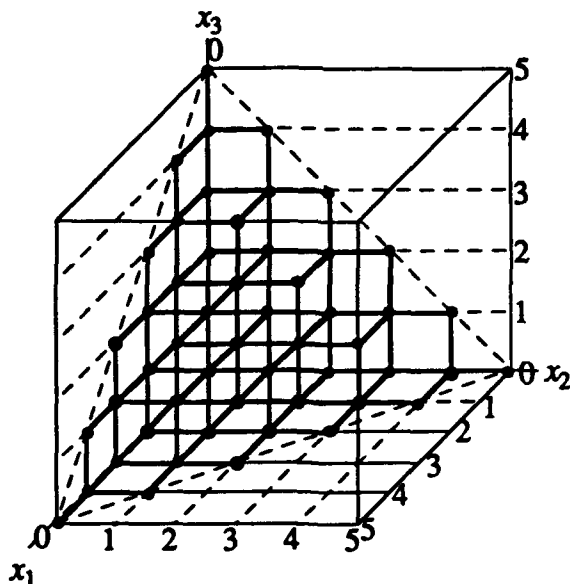


Fig. 2 — The admissible state space Ω for the network of Fig. 1

In general, a vector description of circuit j in terms of the nodes it traverses is given by

$$c_j = \{c_{j1}, c_{j2}, c_{j3}, \dots, c_{jN}\},$$

where

$$c_{ji} = \begin{cases} 1, & \text{if circuit } j \text{ traverses node } i \\ 0, & \text{otherwise} \end{cases},$$

and N is the number of nodes in the network. In the example shown in Fig. 1, circuit 1 is thus represented as

$$c_1 = \{1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0\}.$$

Now we can express the system constraints as

$$\sum_{j=1}^J x_j c_{ji} \leq T_i, \quad i = 1, \dots, N \quad (1)$$

where J is the number of types of circuits in the network. Although the system constraints limit the number of simultaneously active calls at each node, in radio networks there still exists a certain degree of flexibility in how the transceivers at each node are configured. For example, if node 1 has three transceivers, it can configure its connections in a variety of ways (such as all three connections to node 2; one connection to each of nodes 2, 3, and 5, etc.) as long as the total number of simultaneously active calls does not exceed three (and provided that the neighboring nodes also have transceivers available for this purpose).

The situation in wireline networks is similar, but the flexibility to reconfigure a node's connections is not present because the number of channels available on each link connecting a pair of nodes remains constant. In this case a system of equations similar to Eq. (1) can be written in which the index i now refers to a particular link (rather than to a node). Thus c_{ji} would take on the value of 1 if and only if circuit j traverses link i . The system constraints would be again written in terms of T_i , which would be now interpreted as the number of calls that can be supported by link i .

In this report we do not address the protocol issues that are associated with call setup, such as the control messages that must be exchanged to disseminate the network state to all nodes. Our focus is on the development of a mathematical system model that demonstrates the performance improvement that can be achieved through the use of admission control. Protocol issues will be addressed in the future, and will take advantage of the insights gained through these studies.

2.2 Control Policy

Our ultimate goal is to achieve optimal network performance by exercising an admission control policy on calls. In our studies of voice-only networks we use the criterion of blocking probability; when we extend our study to integrated networks we plan to use the

weighted sum of blocking probability and expected data-packet delay. In practice, the true optimal solution is usually elusive, and we must settle for a good suboptimal solution.

The system constraints limit the state space Ω in which x is allowed to take values. We assume that the state space is coordinate convex [15]. The primary characteristic of such systems is that if x is an admissible state ($x \in \Omega$) and $x_j \geq 1$, then $x' = (x_1, x_2, \dots, x_{j-1}, \dots, x_j)$ must also be an admissible state ($x' \in \Omega$). This condition implies the very reasonable property that call completions are not blocked, and that call durations are independent of the system state. We consider policies that retain the coordinate convexity of the state space. Under such policies, a new call is admitted with probability 1 if the state to be entered is in the admissible region; otherwise, it is blocked with probability 1. The objective is to determine the coordinate convex set Ω^* that provides the optimum value of the specified performance index. Thus, a coordinate convex policy is specified in terms of the set of admissible states, which is defined by a set of linear inequality constraints corresponding to coordinate convex regions in discrete state space. The control policy is effectively a further restriction of the admissible state space defined by the system constraints (Eq. (1)). Under reasonable conditions on arrival and service time distributions, the use of coordinate convex policies results in a product-form characterization of the system state, which greatly simplifies the evaluation of system performance, as is discussed in Section 2.3.

Although coordinate convex policies are not necessarily optimal, Jordan and Varaiya have provided examples in which they perform almost as well as dynamic programming solutions [9].⁸ The advantages of coordinate convex policies are that they are easy to implement and evaluate because of their product-form solution, thus permitting the solution of considerably larger problems than are possible by using dynamic programming.

For notational purposes, we subdivide the control policy into a set of circuit "thresholds," and a set of "control constraints." Thresholds restrict the number of calls that will be admitted to the individual circuits, and can be expressed as

$$x_j \leq X_j = \text{threshold on circuit } j, \quad 1 \leq j \leq J. \quad (2)$$

Threshold X_j "cuts" the state space with a hyperplane that is perpendicular to the x_j axis and intersects the axis at X_j , as illustrated in Fig. 3.

The control constraints are restrictions on the sums of the number of calls of various types, i.e.,

⁸ Dynamic programming policies, which are in general not coordinate convex, specify not only a set of admissible states but also which transitions between admissible states are permitted.

$$\sum_{j \in I} \alpha_{Ij} x_j \leq Y_I \quad (3)$$

for various sets I , where I is a subset of the set of all call types. The coefficients α_{Ij} are included in Eq. (3) to permit different weights for different call types. One case that may require the use of nonunit values of α_{Ij} (i.e., $\neq 1$) is a network that offers services which have different bandwidth requirements; e.g., transmission of video may require the equivalent bandwidth of several voice channels. In fact, nonunit values may be required even when all services use the same bandwidth. Unit values of the coefficient α_{Ij} allow control constraints to cut the state space with a "45°" hyperplane. An example in which $I = \{1, 2\}$, $\alpha_{Ij} = 1$ for all j , and $Y_I = 4$ is shown in Fig. 4.

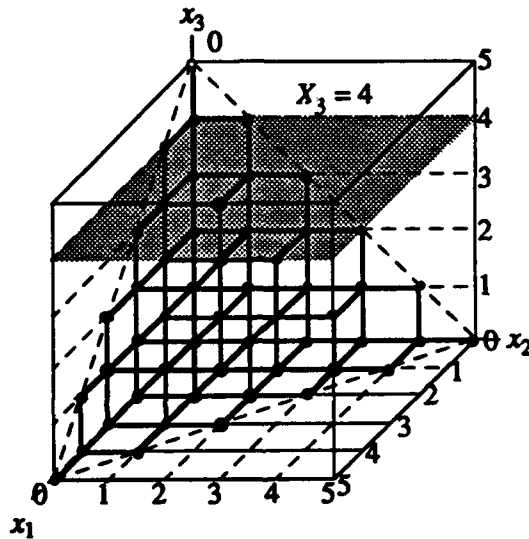


Fig. 3 — State space restricted by setting threshold $X_3 = 4$

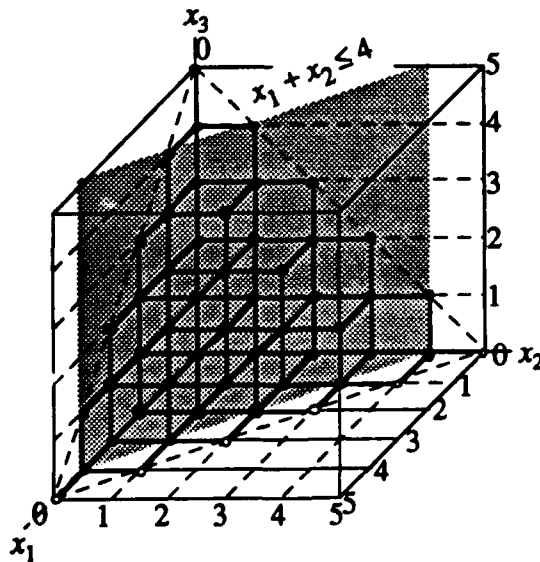


Fig. 4 — Example control constraint

Foschini and Gopinath [16] have shown that, for multiple service single resource (MSSR) networks with 3 services (the present example fits into this class of problems), unity values for the α_{Ij} are sufficient to obtain the optimal coordinate convex control policy. Jordan and Varaiya [9] have generalized this result to include any number of services. However, they have also shown that the values of α_{Ij} might be different from 1 in the optimal coordinate convex policy for MSMR (multiple services multiple resources) networks, which, in general, include the networks we are considering.

Although it is generally necessary to evaluate all coordinate convex policies—that is, all combinations of threshold values, all I 's that represent linear combinations of the circuits with all values of Y_I , and all appropriate values of α_{Ij} —to discover and verify the optimal one, we use $\alpha_{Ij} = 1$ throughout this paper. This simplification is required because of the complexity of the problem for even small networks. Since the parameters we wish to optimize are the boundaries of the state space (which are specified by the control policy), the dimensionality of the problem is equal to the number of control policy constraint equations. (This can be seen in Fig. 4, where the optimum value of Y_I must be determined in addition to the optimal threshold values for the three call types.) Consideration of nonunit α_{Ij} would cause the dimensionality to explode.⁹ The use of $\alpha_{Ij} = 1$ is also partially justified by the theoretical results of Foschini and Gopinath [16] and Jordan and Varaiya [9], and our empirical observations that indicate that control policies using only 45° constraints are very nearly, if not actually, optimal.

An important question is how to organize the description of the different coordinate convex regions. The individual thresholds must be examined, and the sets I must be determined. We find the sets I by examining the set of circuits that intersect at each node. In the network of Fig. 1, the only location that circuits intersect is at node 1 where all three circuits cross. A control constraint $x_1 + x_2 + x_3 \leq Y_I$ is not useful because it amounts to reducing the capacity of node 1 to the Y_I value. Therefore, we only consider the following three control constraints:

$$x_1 + x_2 \leq Y_1,$$

$$x_1 + x_3 \leq Y_2,$$

$$x_2 + x_3 \leq Y_3.$$

We have found that adjusting the thresholds is generally the most powerful form of optimization; once the optimal set of thresholds (based on a threshold-only search) has been

⁹ Note that if the problem were not a discrete one, the dimensionality would actually become infinite. Although there are infinitely many values of α_{Ij} , there are only a finite number of discrete subspaces.

found, continued search over the set of control constraints (i.e., adjusting the values of the Y_I 's) typically yields only small reductions in the blocking probability.

2.3 The Product-Form Distribution

We assume that the generation process for a call of type j (i.e., a call that uses circuit j) is Poisson with arrival rate λ_j , and that its duration is exponentially distributed with mean $1/\mu_j$; the corresponding load is $\rho_j = \lambda_j/\mu_j$. Furthermore, control is centralized, and the resources (at all nodes) needed by a circuit are acquired simultaneously when the call arrives and are released simultaneously when the call is completed. Calls are blocked when one or more nodes along the path do not have a transceiver available or when a decision is made not to accept a call, as is discussed below. Blocked calls are assumed to be lost from the system.¹⁰

Under these conditions, in conjunction with the use of coordinate convex policies, the Markov chain describing the system state is time reversible and has the product-form stationary distribution

$$\pi(x) = \pi(0) \prod_{j=1}^J \frac{\rho_j^{x_j}}{x_j!},$$

where $\pi(0)$ is the normalization constant given by

$$\pi(0) = \left\{ \sum_{x \in \Omega} \prod_{j=1}^J \frac{\rho_j^{x_j}}{x_j!} \right\}^{-1}.$$

It is convenient to define an index quantity associated with any given subset Ω' of the state space by

$$G(\Omega') = \sum_{x \in \Omega'} \prod_{j=1}^J \frac{\rho_j^{x_j}}{x_j!}.$$

Then the connection between the admissible state space Ω and the crucial quantity $\pi(0)$ is provided by the relation $G(\Omega) = \{\pi(0)\}^{-1}$.

The control policy is defined by the specification of the admissible state space Ω . For any such state space, it is straightforward (though time consuming) to evaluate $\pi(0)$, which

¹⁰ It is certainly possible to consider systems in which blocked calls are queued at the source node (instead of being lost) until resources for them become available at all nodes along the path. Although a Markov model can be set up for such a system, the practical applicability of this approach is limited to small systems because the product-form solution does not apply in that case. The equilibrium distributions of the Markov chain can be evaluated either by means of the balance equations or by stepping it through the transitions until it "relaxes" as steady state is approached.

in turn permits the evaluation of performance measures such as throughput and blocking probability. In state x the total number of active calls is

$$\gamma(x) = \sum_{j=1}^J x_j.$$

Throughput $\Gamma(\Omega)$ is simply the expected number of active calls averaged over the system state:

$$\Gamma(\Omega) = \sum_{x \in \Omega} \{\gamma(x)\pi(x)\}.$$

Blocking probability $P_b(\Omega)$ is the ratio of the expected number of blocked calls per unit time to the expected total number of call arrivals per unit time:

$$P_b(\Omega) = \frac{\sum_{j=1}^J \rho_j P_{bj}(\Omega)}{\sum_{j=1}^J \rho_j} = \frac{\sum_{(x: \text{blocking state})} \sum_{j=1}^J 1((x_j + 1) \notin \Omega) \rho_j \pi(x)}{\sum_{j=1}^J \rho_j},$$

where $P_{bj}(\Omega)$ is the probability of blocking a call of type j (i.e., the fraction of type- j calls that are blocked), a “blocking state” is any one of the states on the boundary of the region Ω in which at least one type of call must be blocked, and $1(\cdot)$ is the indicator function, which is 1 if the argument is true and 0 otherwise.

Thus the goal is to restrict the admissible state space to a coordinate convex region such that the desired performance measure is optimized. The direct approach is to compute $\pi(0)$ and the performance index for all possible coordinate convex regions. The discrete optimization procedure developed by Jordan and Varaiya [9] to make the search more efficient is not directly applicable when the performance metric is blocking probability because their procedure requires that the cost criterion be a function of the state and not dependent on the control policy.¹¹ They have also developed a continuous model that reduces complexity [17, 18], but this model, like the discrete one, is more appropriate for throughput, which is indeed directly a function of the system state. In our paper we have developed a recursive procedure to accelerate the evaluation of a large number of different admission-control

¹¹ To see that this is true, consider the performance measure associated with each state x (which is then averaged over the admissible set of states to obtain system performance). The performance criterion of blocking probability has the property that it has a value of 1 for calls of type j in states that are blocking states for calls of type j , and zero elsewhere. Thus its value depends on the control policy. In contrast, the criterion of throughput has a value of $\gamma(x)$ in state x , independent of the control policy. However, optimizing throughput is equivalent to optimizing blocking probability. Therefore, Jordan and Varaiya's approach can be indirectly used to minimize blocking probability.

policies, and a descent-search method to minimize the number of policies that must be evaluated in searching for the optimal one; both of these are applicable to either cost criterion.

3 COMPUTATIONAL ISSUES IN EVALUATING PRODUCT-FORM SOLUTIONS

Knowledge that the equilibrium distribution $\pi(x)$ satisfies the product-form distribution greatly simplifies the evaluation of system performance. Since the distribution is known to within the normalization constant $\pi(0)$, there is no need to solve the balance equations associated with the Markov chain that describes our system. However, the evaluation of $\pi(0)$ is computationally intensive because it requires the evaluation and summation of a large number of terms of the form $\prod \rho_j^{x_j} / x_j!$. Considerable effort has been exerted in developing efficient procedures for calculating the normalization constant (see e.g., [19]), but such methods are generally problem specific and of little help to our problem. In this section we discuss the approaches we have used.

3.1 A Recursive Procedure

Recursive techniques are used in many of the schemes that have been proposed to evaluate the normalization constant. Since we must calculate $\pi(0)$ (equivalently, $G(\Omega')$) for a potentially large number of subspaces in our problem, we have developed a different recursive technique that exploits previous calculations of $G(\Omega)$ for different Ω , which are subsets of Ω' .

Figure 5 shows the basic principle of our approach by means of a two-dimensional example. Assume we have already evaluated the normalization constant associated with region Ω , and now want to evaluate the normalization constant associated with region Ω' . The brute-force approach would be simply to add up all the terms of the form $\prod \rho_j^{x_j} / x_j!$ that correspond to policies in Ω' . However, it is easy to see that Ω' contains all of the policies in Ω , plus a few additional ones. Thus $G(\Omega') = G(\Omega) + G(\Omega' - \Omega)$. Therefore, the only policies for which terms of the form $\prod \rho_j^{x_j} / x_j!$ have to be computed are those in the region $\Omega' - \Omega$; their sum is added to the already-available quantity $G(\Omega)$, thereby resulting in a significant reduction in the computational burden. Actually the computation process is still somewhat complex because the sequence of coordinate convex regions that are examined is not monotonically increasing in all dimensions, and consequently it is necessary to keep track of which of the previously examined regions provides the most useful information for the evaluation of the normalization constant corresponding to the subspace of current interest. Doing so requires a great deal of bookkeeping. Even so, considerable improvement is achieved as compared with the direct approach.

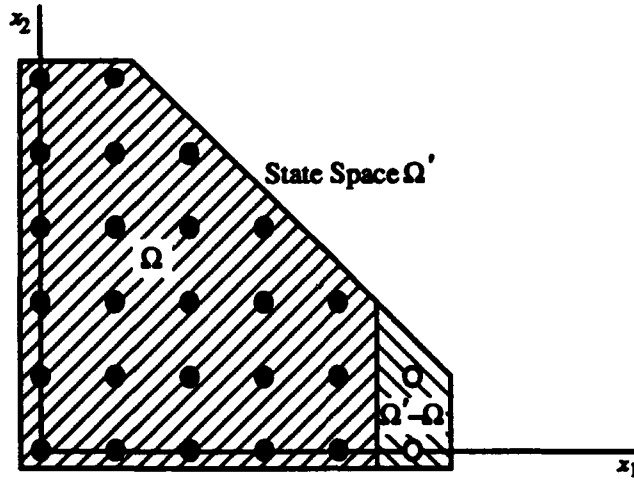


Fig. 5 — A two-dimensional example of our recursive procedure

Thus our recursive procedure facilitates the search of the multitude of coordinate convex regions by reusing the frequently performed calculations for smaller regions in calculating a performance index for a larger region. For example, let $P_b(\Omega)$ denote the blocking probability for a coordinate convex region Ω , B denote the set of blocking states for this region, and $B_j \subset B$ denote the set of states that block calls of type j . If we have computed $P_b(\Omega)$ then we can compute $P_b(\Omega')$, for a region Ω' that is created by suitably augmenting Ω , as follows:

$$P_b(\Omega') = \frac{\sum_{j=1}^J \rho_j P_{bj}(\Omega')}{\sum_{j=1}^J \rho_j} = \frac{\sum_{j=1}^J \rho_j G(B'_j)}{G(\Omega') \sum_{j=1}^J \rho_j}$$

$$= \frac{\sum_{j=1}^J \rho_j G(B_j) + \sum_{j=1}^J \rho_j G(B'_j - B_j) - \sum_{x \in B} \left(\prod_{i=1}^J \frac{\rho_i^{x_i}}{x_i!} \sum_{j=1}^J 1((x_j + 1) \notin \Omega, (x_j + 1) \in \Omega') \rho_j \right)}{(G(\Omega) + G(\Omega' - \Omega)) \sum_{j=1}^J \rho_j},$$

where both the first expression in the numerator and $G(\Omega)$ have been previously calculated. Using this recursive formula to evaluate different regions Ω' , the tedious calculation of the blocking probability for many regions is essentially reduced to simply summing the probabilities that the system is in one of the states in the slice of space $\{\Omega' - \Omega\}$.

Thus, we are able to use data obtained in performing the relatively inexpensive calculations of the normalization constant for the smallest state spaces to efficiently calculate the normalization constant for larger state spaces. Since we cannot know a priori which state

space will provide the best performance, they all must be evaluated.¹² Although our approach differs from the recursions described in [19], the two methods are not mutually exclusive. We believe that our method could be made more efficient by the inclusion of techniques similar to those described in [19].

3.2 Descent Search and the Question of Unimodality

We noted earlier that the direct method of determining the optimal policy is to evaluate performance for all possible coordinate convex regions, an enormous computational task for all but very small systems. Thus, as other researchers have also observed, it is often advisable (or even mandatory) to use a non-exhaustive search strategy that attempts to find a good, although generally suboptimal, policy. However, in systems with a single local optimum of the performance measure, which is also the global optimum, gradient or descent-search methods provide a non-exhaustive search strategy that can be used to expedite the search of the state space without sacrificing achieving the optimal solution. Performance measures that have a single local optimum are known as unimodal. Although we have not rigorously verified that blocking probability is indeed a unimodal function of the coordinate convex policy parameters for the type of network we are considering, empirical evidence, which is presented here and in Section 4.1, suggests that it is. Empirical evidence also indicates that, even if the unimodality conjecture is not correct, descent search methods are effective heuristics for finding good admission control policies. In this subsection we address the issue of unimodality and how it may be used to enhance the search for the optimum policy.

First we consider a very simple example with only one circuit, which is operating at a utilization rate of $\rho = 1$. Figure 6 shows the blocking probability P_b as a function of the threshold X_1 . As one would expect, P_b decreases monotonically with X_1 . Thus the unimodal property is satisfied. A slightly more complicated example is shown in Fig. 7, which represents the case in which two circuits, each with $\rho = 3$, share a node and hence compete for use of network resources. The total capacity of the node is ten simultaneous calls. Again, the behavior is rather uninteresting because the optimum policy is to admit all calls (of either type) until the system is saturated. Thus unimodality holds in this case also.

¹² Except when using the descent-search technique described in Section 3.2. However, our recursive procedure is still helpful in this case, even though conducting a descent search does not require evaluation of all subspaces.

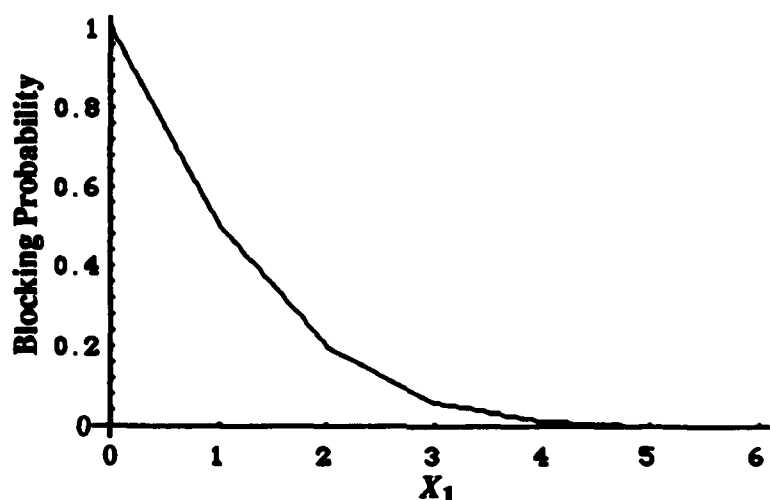


Fig. 6 — Blocking probability for a single circuit as a function of the threshold X_1 , $\rho = 1$

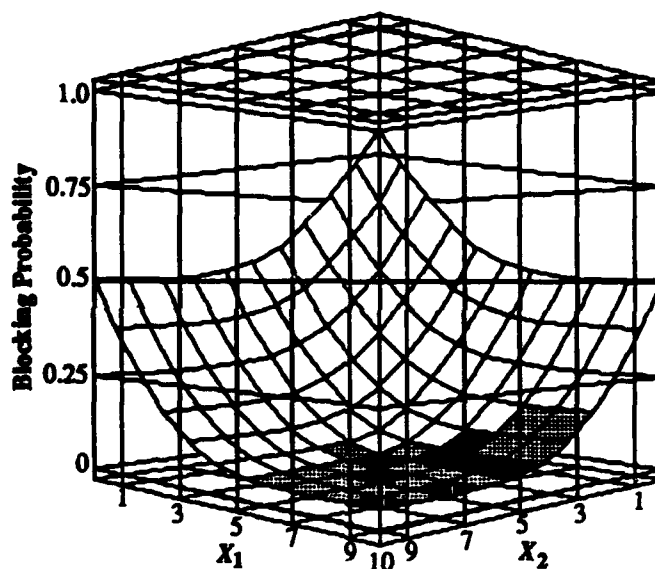


Fig. 7 — Blocking probability as a function of the thresholds X_1 and X_2 ($\in \{0,1,2,\dots,10\}$) for a pair of circuits that intersect at a node, $\rho = 3$

This problem can be made more interesting by associating different costs with the blocking of calls of different types. Again we consider a system with two call types sharing a single node. However, we now associate different costs with blocking calls of the different types. In particular, the cost of blocking type-1 calls is 2 and that of blocking type-2 calls is 1. Figure 8 shows the “blocking cost” for such a system with $\rho = 20$ and a nodal capacity of 10 calls (a very heavily loaded system); two views are shown to permit a better understanding of the 3-dimensional surface. In this case, the optimal policy is no longer to

admit all calls. The minimum value of blocking cost is obtained by admitting only type-2 calls.¹³ However, system behavior is again unimodal.

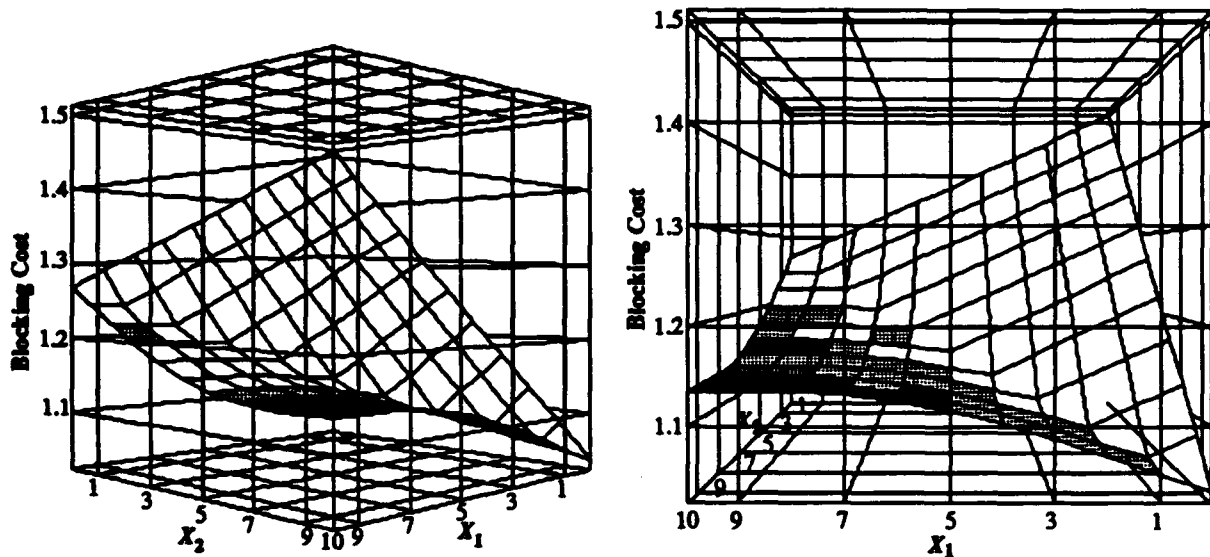


Fig. 8 — Blocking “cost” as a function of the thresholds X_1 and X_2 ($\in \{0,1,2,\dots,10\}$) for a pair of circuits that intersect at a node, $\rho = 20$, cost of blocking type 1 calls = 2, cost of blocking type 2 calls = 1

Although the figures related to these specific examples demonstrate unimodality, we have not been able to prove that blocking probability is a unimodal function of the boundaries of the state space. Nonetheless, the figures do suggest that using some form of a steepest descent or gradient search method may prove helpful in our search for the optimal control policy. Furthermore, if the conjecture of unimodality can be substantiated, we are guaranteed to converge to an optimal control policy when we use a descent-based search method.

3.2.1 Descent Search

Gradient search is normally associated with continuous space; the literature contains much research into and many variations of gradient search in the continuous domain. A good overview and several examples are presented in [20]. In the discrete domain, however, to speak of gradient search is not technically correct because derivatives do not exist. Instead, we use a heuristic descent-search method to perform an efficient and effective search for the optimal control policy. In this section we present an overview of the search technique; a detailed description and an example of our descent search method are given in Section 5.1.1.

Recall that the variables we wish to optimize, the X_j 's and the Y_j 's, actually define the boundaries of the state space representing the number of calls of each type. The search

¹³ At lower values of ρ some calls of type 1 are admitted.

starts from some policy $\Omega_0 = \{X_1, \dots, X_J, Y_1, \dots, Y_I\}$, which is typically (but not necessarily) the uncontrolled state space where the X_j 's and the Y_I 's are all set at their maximum value. The idea is to search among immediate neighbors of Ω_0 (i.e., control policies that only vary from the policy Ω_0 by ± 1 in a few coordinates¹⁴) for a policy that gives better performance. If improved performance is possible at one of the neighboring policies, say Ω , move to that point (i.e., set $\Omega_0 = \Omega$) and, again, search among immediate neighbors of Ω_0 for a policy that gives better performance. Continue in this manner until the search among neighbors finds none that can provide better performance. If the conjecture of unimodality is correct, this point is the optimal control policy; otherwise, it is only a local minimum that may or may not be the optimal policy. To verify the unimodality property a bit more conclusively, although without total certainty, we have performed descent searches from many different initial policies to see if different solutions are found. In every case examined, the searches converged to the same solution.

We perform the search in two loops. In the first loop, which we call the thresholds-only loop, the optimum "threshold-only" policy (i.e., the optimum set of threshold values $\{X_j\}$) is found (the control constraint limits [the Y_I] are set at their maximum value). A descent search of threshold-only policies can generally be rapidly performed. It has been our experience that such a search yields a control policy that is nearly optimal. Thus, the threshold-only loop efficiently moves the search close to the optimum. The second loop, called the "combined loop," continues the search from the optimal threshold-only policy; in this loop both the thresholds and the control constraint limits are optimized.¹⁵ Because this loop includes both the thresholds and the control constraints, it can be much more time consuming than the thresholds-only loop.

4 NUMERICAL EVALUATION OF NETWORK PERFORMANCE

The examples presented in this paper were evaluated using computer programs that were developed and refined using Mathematica [21]. Mathematica provides a method for performing symbolic as well as numeric evaluation of tedious expressions. However, symbolic evaluation rapidly becomes unwieldy, both in terms of time and memory, as the number of symbolic variables grows. Therefore, we inevitably end up performing numeric evaluation. Many of our numeric analyses have been run on Macintosh computers using a Mathematica program. The more costly runs (in terms of time and memory) have been programmed in C and C++ and run on Sun 4 and IBM RISC system 6000 workstations.

¹⁴ Changing several coordinates simultaneously results in "diagonal" moves to neighboring policies.

¹⁵ The thresholds determined in the threshold-only loop may be changed as the search proceeds through the combined loop.

4.1 A 10-Node, 5-Circuit Example

Many of our numerical examples are based on the sample network shown in Fig. 9, in which links are shown connecting all nodes that are within communication range of each other. Figure 10 shows five circuits superimposed on this network graph; a call on one of these circuits requires the use of one transceiver at every node along the corresponding path, as discussed earlier. We can view this network as providing five services, one corresponding to each of the five circuits.

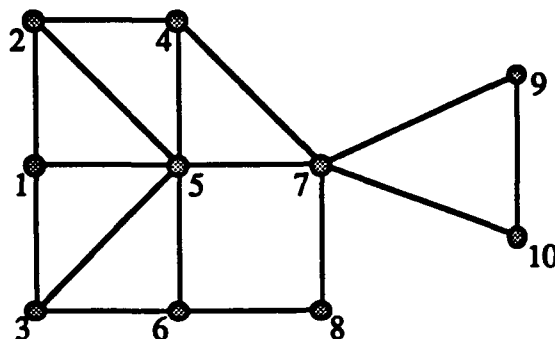


Fig. 9 — An example ten-node multihop network

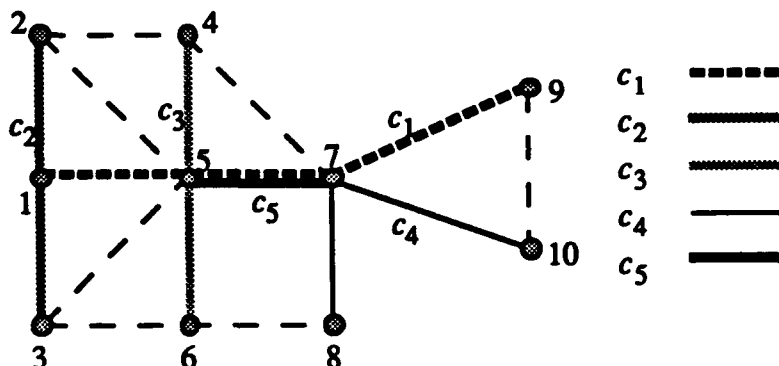


Fig. 10 — Five circuits superimposed on the network of Fig. 9

By examining Fig. 10, we can expand the system constraints Eq. (1) as follows:

$$x_j \leq \min_{i \in \text{node } i \in \text{circuit } j} \{T_i\}, \quad j = 1, \dots, 5, \quad i = 1, \dots, 10$$

$$x_1 + x_2 \leq T_1 \quad (\text{system constraint from node 1})$$

$$x_1 + x_3 + x_5 \leq T_5 \quad (\text{system constraint from node 5})$$

$$x_1 + x_4 + x_5 \leq T_7 \quad (\text{system constraint from node 7}) \quad (4)$$

The first of these inequalities states simply that the number of calls of type j cannot exceed the number of transceivers at any of the nodes along the path. The other three inequalities reflect the fact that the transceivers at a node are shared among the different types of circuit that pass through the node; in this example, only nodes 1, 5, and 7 support more than one

type of circuit. The control policy, which is expressed in terms of the variables X_j and Y_l whose values are to be optimized, is obtained from the system constraints and is written as

$$\begin{array}{ll}
 x_j \leq X_j & \text{(thresholds)} \\
 x_1 + x_3 \leq Y_1 & \text{(control constraints from node 5)} \\
 x_1 + x_4 \leq Y_2 & \text{(control constraints from node 7)} \\
 x_1 + x_5 \leq Y_3 & \text{(control constraints from node 5)} \\
 x_3 + x_5 \leq Y_4 & \text{(control constraints from node 5)} \\
 x_4 + x_5 \leq Y_5 & \text{(control constraints from node 7)}
 \end{array} \tag{5}$$

We consider an example in which the loads of all circuits are equal (i.e., $\rho_j = \rho, j = 1, \dots, 5$), and three transceivers are allocated to each node (i.e., $T_i = 3, i = 1, \dots, 10$). Table 1 shows the optimal¹⁶ policy in terms of overall call blocking probability as ρ is increased from 0.5 to 10,¹⁷ as well as the uncontrolled and the optimal blocking probability P_b .¹⁸ At low network loads, it is best to accept all calls. As ρ is increased, calls of type 1 are the first to have their threshold reduced. This is intuitively satisfying because circuit 1 shares at least one node with each of the other circuits. Calls of type 5 are the next to have their threshold reduced. This is because they interfere with calls of type 3 and 4. Calls of type 2, 3, and 4 never have their thresholds reduced because (after type 1 and type 5 calls have been eliminated) they do not share resources with any other type of call.

It turns out that, for this example, the control constraints are not effective in improving performance beyond the improvement provided by the thresholds alone. Although Y_3 does, in fact, decrease with increasing ρ , thereby suggesting that it has an impact on system performance, examination of Table 1 shows that this variation is simply a tracking of the quantity $X_1 + X_5$, and thus provides no new information. Holding $Y_3 = 3$ constant for all listed values of ρ and optimizing only over the threshold values yields the optimal P_b value. Conversely however, holding the X_j values constant at 3 and optimizing over the Y_l only

¹⁶ We use the term optimal rather casually in this paper. As a result of the complexity of this problem, we have not performed an exhaustive search of all coordinate convex policies for any of the examples considered. Therefore, we cannot be certain that any solution we find is truly optimal. However, we conjecture that these solutions are optimal because the use of a descent search has not found any better solutions. This constitutes sufficient evidence provided that blocking probability is actually a unimodal function of the boundaries of the state space as discussed in Section 3.2.

¹⁷ Values of ρ greater than 1 can be acceptable because node i has T_i (typically greater than one) transceivers. Actually, there is no fundamental limit on ρ , which represents the offered traffic; however, it is clear that P_b approaches 1 for sufficiently large ρ .

¹⁸ Table A1 in the Appendix presents the policies found for the same network and parameters, but based on throughput as the performance measure. As that table shows, optimizing over blocking probability or throughput yields the same control policy.

yields the optimal P_b value for $\rho \geq 3.0$. Of course, for the large ρ values $Y_3 = 0$ is equivalent to $X_1 = X_5 = 0$.

Table 1 — Optimal coordinate convex control policies for the network of Fig. 10

ρ	Thresholds					Control Constraints					uncontrolled	optimal	%† gain
	X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4	Y_5	P_b	P_b	
0.5	3	3	3	3	3	3	3	3	3	3	0.13528	0.13528	0.0
1.0	2	3	3	3	3	3	3	3	3	3	0.32221	0.32203	0.05493
1.5	0	3	3	3	3	3	3	3	3	3	0.44626	0.43965	1.48052
2.0	0	3	3	3	2	3	3	2	3	3	0.52972	0.51220	3.30760
2.5	0	3	3	3	1	3	3	1	3	3	0.58958	0.56694	3.83968
3.0	0	3	3	3	0	3	3	0	3	3	0.63485	0.60769	4.27846
10	0	3	3	3	0	3	3	0	3	3	0.85086	0.83924	1.36545

† % gain = $100 (P_b(\text{uncontrolled}) - P_b(\text{optimal})) / P_b(\text{uncontrolled})$

The last column in Table 1 shows the “% gain,” i.e., the percentage of reduction in the call blocking probability obtained by restricting the state space. The maximum improvement found was 4.27% when $\rho = 3$. This raises a fundamental question: Is the improvement obtained by administering control worth the effort invested? This question cannot be answered on the basis of a single isolated example, but it is worth considering as our studies progress. Many of the examples presented in this report were motivated by the desire to find a case where the use of an admission control policy has a substantial impact on the network performance. Tables A2 and A3 in the Appendix are examples in which different ρ values for each circuit were examined with the thought that the presence of circuits with high (or low) utilization rates (ρ values) relative to the other circuits might allow a control policy to have a more substantial impact than is shown in Table 1. However, Tables A2 and A3 indicate that various utilization rates do not appreciably effect the gain obtained by administering control. In Section 5.2 we explain this inability to obtain significant performance gains by showing how networks tend to be “self regulating.”

The optimal policies listed in Table 1 were obtained by performing an exhaustive search over all subspaces defined by the thresholds and control constraints of Eq. (5). This search was facilitated by our recursive technique for computing the normalization constants and the corresponding blocking probabilities associated with each candidate subspace. It is instructive to observe the behavior of the blocking probability as the admissible subspace is varied. These observations may give some insight into the conjecture that blocking probability is a unimodal function of the state space boundaries.

Figures 11 - 13 show the effects of a call admission policy in which *only* the thresholds are adjusted to minimize call blocking probability. The axes on these curves are arranged in a rather nonstandard way, which requires explanation. The z axis shows the

Thresholds for circuit 1 and 2 calls

Thresholds for circuit 3, 4, and 5 calls

The minimum blocking probability is obtained with thresholds (3,3,3,3,3)

Figure 11 indicates that when the traffic levels are low ($\rho = 0.5$) the best admission policy is to accept calls whenever the required resource is available, i.e., the optimal set of call thresholds is $\{3,3,3,3,3\}$. However, as the traffic levels increase, the blocking probability can be reduced by limiting the number of calls on certain circuits. Figure 12 shows that with $\rho = 2.0$ for each circuit, the blocking probability is minimized by blocking all calls on circuit 1 and allowing no more than two calls on circuit 5. Figure 13 shows that when $\rho = 3.0$, the optimal admission policy is to accept calls only on circuits 2, 3, and 4, and to block all calls on circuits 1 and 5.

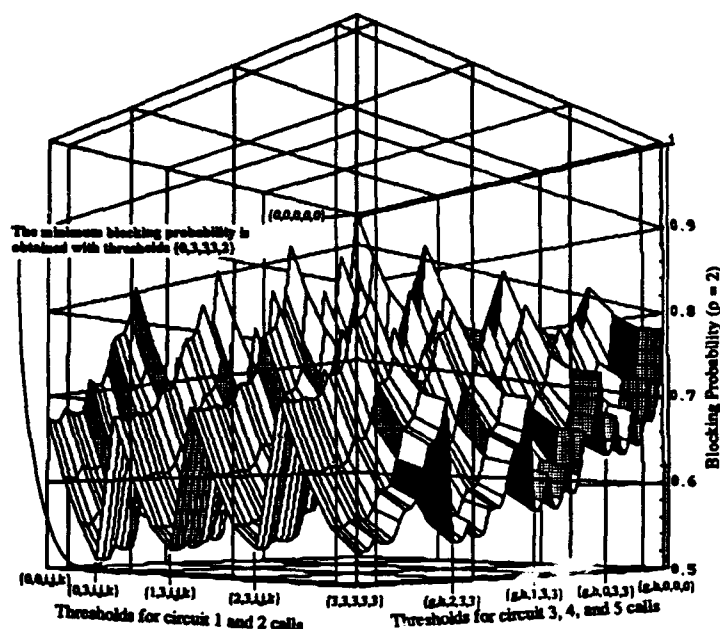


Fig. 12 — Blocking probability vs. threshold values for the five circuits of Fig. 10, $\rho = 2.0$

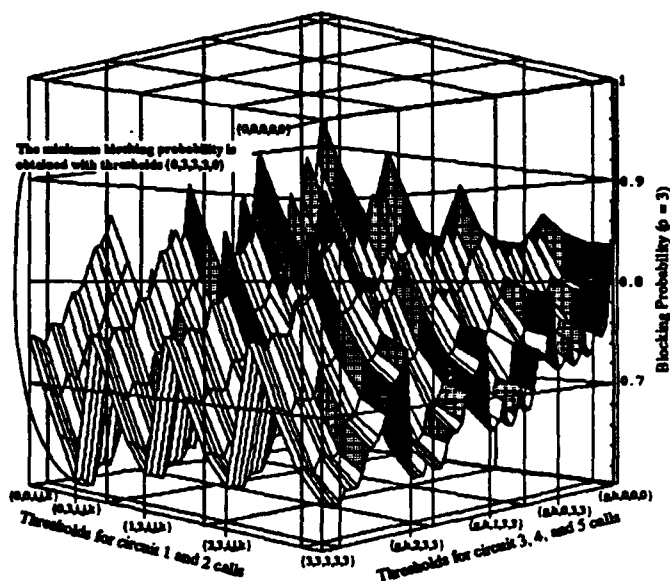


Fig. 13 — Blocking probability vs. threshold values for the five circuits of Fig. 10, $\rho = 3.0$

The Unimodality Issue Revisited

Although the plots of Figs. 11 - 13 do not at first appear to exhibit the property of unimodality, upon closer examination it can be seen that the figures actually reinforce the unimodality conjecture. The initial multimodal appearance is a result of the organization of the threshold values along the axes. We are forced to view a six-dimensional plot in three dimensions. When the organization of the data is understood, one can see that the ridges are

the result of a "dimensional shift;" properties of unimodality are exhibited almost everywhere, except at the points of transition, e.g., from $X_j = 3$ to $X_j = 0$.

To clarify this point, consider the plots shown in Figs. 14 and 15. In these figures only the thresholds X_1 and X_5 are varied through the range $[0, 6]$. The remaining thresholds X_2 , X_3 , and X_4 , as well as the Y_i are held constant at the value $T_i = 6$. Thus, we are able to present a legitimate three-dimensional plot of blocking probability versus thresholds. In the figures, additional plots (14(b), 14(c), and 15(b)) are used to magnify the ordinate about the minimum blocking probability value; these plots show that there is, in fact, a single local optimum. Figure 14 shows that when $\rho = 2.0$, the optimal threshold settings are $X_1 = 3$ and $X_5 = 6$. When $\rho = 3.5$, as in Fig. 15, the optimal threshold settings are $X_1 = 0$ and $X_5 = 3$.

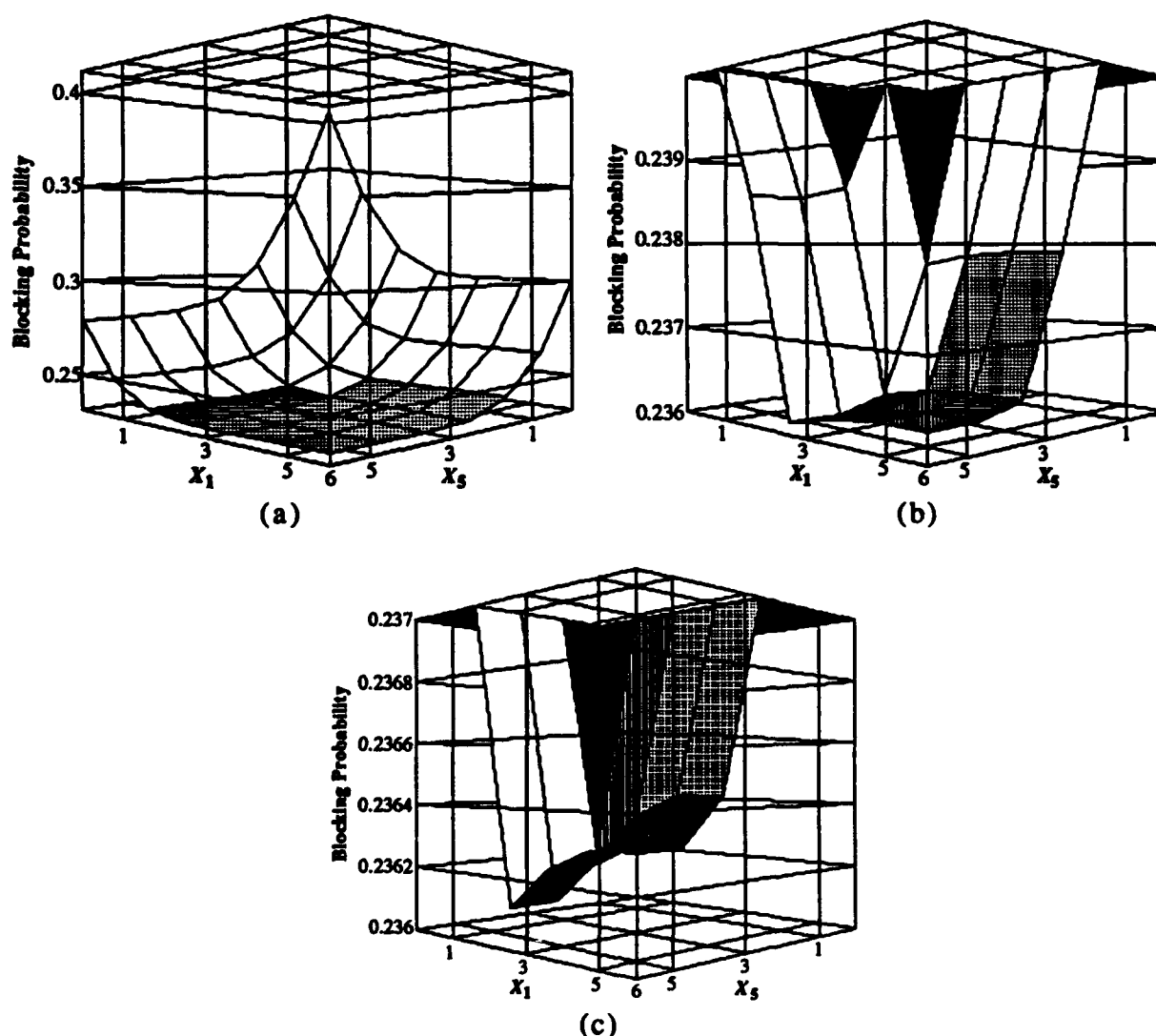


Fig. 14 — Plots of blocking probability vs. threshold values that illustrate unimodality, $\rho = 2.0$

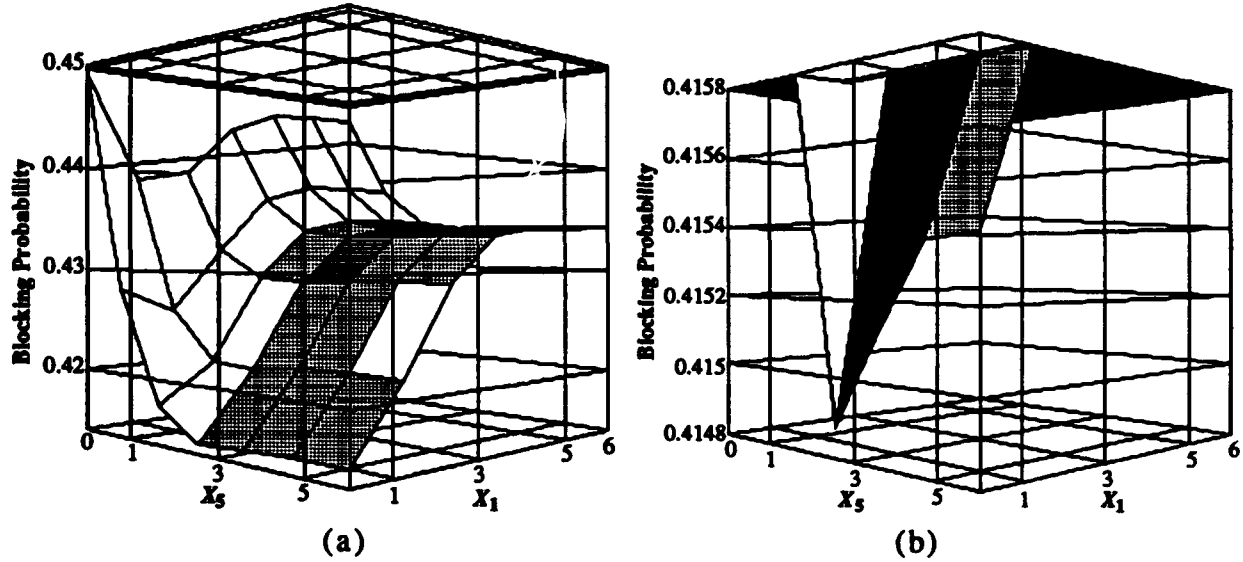


Fig. 15 — Plots of blocking probability vs. threshold values that illustrate unimodality, $\rho = 3.5$

4.1.1 Blocking Cost Minimization: A Weighted Performance Index

A commonly used approach in problems of this type is to attempt to optimize a cost function related to throughput or blocking probability, rather than optimizing the overall network throughput or blocking probability. Thus a “weight,” which reflects the revenue returned for providing the service, or the cost (e.g., lost revenue) of failing to provide the service (i.e., blocking the call), is assigned to each call type. For example, certain circuits may be more heavily weighted because they carry traffic of higher priority. This approach is reasonable for the network of Fig. 10, because circuit 1 clearly uses more resources, and creates more conflicts, than any of the other circuits. If the revenue returned for providing circuit 1 is not greater than that for the other circuits, it would probably be most cost effective to eliminate circuit 1. This is, in fact, what the analysis of the previous section has suggested (i.e., as the traffic rate increases, the resources allocated to circuit 1 are the first to be reduced).

We have examined the effect of weighting in the network of Fig. 10. Calls on circuits 2, 3, 4, and 5 were assigned an equal weight of $w_{2+} = 1$, and different weighting values w_1 were evaluated for calls on circuit 1. These weights were incorporated into a “blocking cost” performance measure defined by

$$P_b^w = \frac{\sum_{j=1}^5 \rho_j w_j P_{bj}}{\sum_{j=1}^5 \rho_j},$$

where P_{bj} is the probability of blocking a call on circuit j , i.e., the fraction of type- j calls that are blocked. The results of descent searches based on this metric and four different weighting factors are shown in Table 2 (only the thresholds are shown because the control constraints did not affect the results). With type 1 calls 1.5 times as costly as the other calls, the optimal control policy is the same as that for the unweighted network. As w_1 increases, however, the optimal policy shifts to favor circuit 1 calls so that when $w_1 \geq 5$ the only calls admitted to the network are those on circuit 1. With this weighting scheme, the gain obtained by administering control can be made arbitrarily large by increasing the call cost.

Table 2 — Optimal control policies for the weighted network of Fig. 10

ρ	w_1	w_{2+}	Thresholds					optimal P_b^w	uncont. P_b^w	%† gain
			X_1	X_2	X_3	X_4	X_5			
3	1.5	1	0	3	3	3	0	0.707692	0.721298	1.88632
3	2	1	2	3	3	3	0	0.796118	0.807741	1.43895
3	5	1	3	0	0	0	0	1.14615	1.3264	13.5894
3	10	1	3	0	0	0	0	1.49231	2.19084	31.8841

$$\dagger \% \text{ gain} = 100 (P_b^w(\text{uncontrolled}) - P_b^w(\text{optimal})) / P_b^w(\text{uncontrolled})$$

4.1.2 Increased Capacity

Thus far, we have not shown a need for the control constraints. Although they were included in the descent search, they were never found to yield a performance improvement in the network of Fig. 10 with three transceivers per node. In this section we let each node have eight transceivers ($T_i = 8$), and restrict the maximum number of calls on any one circuit to six ($X_j \leq 6$). We return to the case in which the calls are not weighted. In Tables 3 and A4 (in the Appendix), we show that the increased capacity allows the control constraints to contribute to the network performance. In Table 3, we introduce the following notation:

- Γ_Ω = the throughput of the uncontrolled system,
- Γ_X = the throughput of the system controlled by adjusting only the thresholds,
- Γ^* = the throughput of the fully controlled (using both thresholds and control constraints) system,
- $P_b(\Omega)$ = the blocking probability of the uncontrolled system,
- $P_b(\Omega_X)$ = the blocking probability of the system controlled by adjusting only the thresholds,
- $P_b(\Omega^*)$ = the blocking probability of the fully controlled system,
- Ω_X = the best control policy for the system controlled by adjusting only the thresholds,
- Ω^* = the optimal control policy (using both thresholds and control constraints).

The threshold policy Ω_X is simply the set of threshold values $\{X_1, \dots, X_5\}$, and the optimal policy is the set of X_j 's and the Y_I 's $\{X_1, \dots, X_5, Y_1, \dots, Y_5\}$. Elements of Ω^* that are different from those corresponding elements of Ω_X are shown in bold. We see that

incorporation of the control constraints permits the increase of some of the threshold values. For example, with $\rho = 3.5$ the threshold X_1 can be increased from 2 to 3 by administering the policy Ω^* , in which the control constraint $(x_1 + x_5 \leq) Y_3$ is set at 5. Thus the maximum number of type-1 calls is increased, but the sum of the number of type-1 and type-5 calls that can be simultaneously active is reduced from 7 to 5 under Ω^* . In this example, use of Ω^* resulted in a slight improvement over the performance obtained by using Ω_X .

Table 3 — Optimal control policies for the network of Fig. 10 with increased capacity ($T_i = 8, X_i \leq 6$)

ρ	Γ_Ω	Γ_X (%gain) [†]	Γ^* (%gain) [†]	$P_b(\Omega)$	$P_b(\Omega_X)$ (%gain) [†]	$P_b(\Omega^*)$ (%gain) [†]	Ω_X	Ω^*
2.5	10.1807	10.1811 {0.0040}	10.1811 {0.0042}	0.18555	0.185512 {0.0175}	0.185511 {0.0183}	{5,6,6,6,6}	{5,6,6,6,6, 8,8,7,8,8}
3.5	12.0336	12.07499 {0.3438}	12.07667 {0.3578}	0.31237	0.310001 {0.7568}	0.309905 {0.7876}	{2,6,6,6,5}	{3,6,6,6,5, 8,8,5,8,8}
4.5	13.2986	13.48778 {1.4222}	13.50657 {1.5635}	0.40895	0.400543 {2.0555}	0.399708 {2.2598}	{2,6,6,6,3}	{2,6,6,6,4, 8,8,4,8,8}
5.5	14.2267	14.57842 {2.4721}	14.62406 {2.7929}	0.48267	0.469876 {2.6497}	0.468216 {2.9935}	{1,6,6,6,2}	{2,6,6,6,3, 8,8,3,8,8}
6.5	14.9364	15.45569 {3.4768}	15.51528 {3.8757}	0.54042	0.524440 {2.9568}	0.522607 {3.2960}	{1,6,6,6,2}	{2,6,6,6,2, 8,8,2,8,8}
7.5	15.4950	16.13004 {4.0985}	16.23304 {4.7632}	0.58680	0.569865 {2.8860}	0.567119 {3.3541}	{1,6,6,6,1}	{2,6,6,6,2, 8,8,2,8,8}
8.5	15.9449	16.68042 {4.6127}	16.77403 {5.1997}	0.62483	0.607519 {2.7697}	0.605317 {3.1222}	{1,6,6,6,1}	{2,6,6,6,2, 8,8,2,8,8}
10	16.4753	17.28274 {4.9006}	17.36501 {5.4000}	0.67049	0.654345 {2.4084}	0.652700 {2.6537}	{1,6,6,6,1}	{2,6,6,6,2, 8,8,2,8,8}
15	17.5489	18.34001 {4.5078}	18.39848 {4.8420}	0.76601	0.755467 {1.3770}	0.754687 {1.4787}	{1,6,6,6,1}	{2,6,6,6,2, 8,8,2,8,8}

[†] All gains are relative to the uncontrolled system.

4.1.3 Increased Congestion

The network of Fig. 10 has the property that circuits 2, 3, and 4 do not share any resources among themselves. In an effort to study systems in which there is a greater degree of resource sharing, we have modified this network as shown in Fig. 16. We have added a new node, node 11, and we have extended circuits 2, 3, and 4 so that they all include this new node, thereby producing a direct competition for system resources among the three circuits. This adds three new control constraints to those enumerated in Eq. (5), and a new system constraint is added to those enumerated in Eq. (4). The complete set of system constraints is now

$$x_j \leq \min_{i \in \text{node } i \in \text{circuit } j} \{T_i\}, \quad j = 1, \dots, 5, \quad i = 1, \dots, 11$$

$$\begin{aligned}
x_1 + x_2 &\leq T_1 && \text{(system constraint from node 1),} \\
x_1 + x_3 + x_5 &\leq T_5 && \text{(system constraint from node 5),} \\
x_1 + x_4 + x_5 &\leq T_7 && \text{(system constraint from node 7),} \\
x_2 + x_3 + x_4 &\leq T_{11} && \text{(new system constraint from node 11).}
\end{aligned} \tag{6}$$

The complete control policy is

$$\begin{aligned}
x_j &\leq X_j && \text{(thresholds),} \\
x_1 + x_3 &\leq Y_1 && \text{(control constraint from node 5),} \\
x_1 + x_4 &\leq Y_2 && \text{(control constraint from node 7),} \\
x_1 + x_5 &\leq Y_3 && \text{(control constraint from node 5),} \\
x_3 + x_5 &\leq Y_4 && \text{(control constraint from node 5),} \\
x_4 + x_5 &\leq Y_5 && \text{(control constraint from node 7),} \\
x_2 + x_3 &\leq Y_6 && \text{(control constraint from node 11),} \\
x_2 + x_4 &\leq Y_7 && \text{(control constraint from node 11),} \\
x_3 + x_4 &\leq Y_8 && \text{(control constraint from node 11).}
\end{aligned} \tag{7}$$

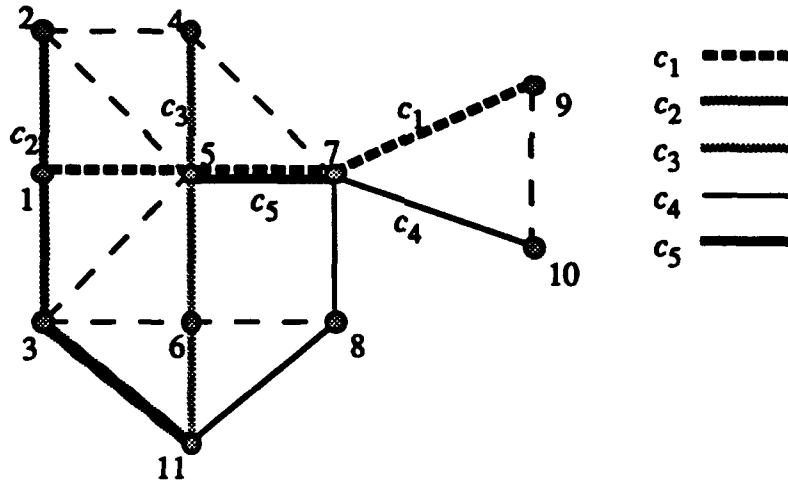


Fig. 16 — A modified, increased-congestion version of the network in Fig. 10

The results of a series of descent searches for the optimal control policy are shown in Table 4. Note that, although improved performance was obtained both by adjusting the thresholds and by exercising the control constraints, the reduction in blocking probability is never more than 2% over a wide range of ρ values. It is interesting to note that the threshold constraints (the first five elements) of Ω^* are identical to the corresponding elements of Ω_X over the entire range of ρ .

Table 4 — Control policies for the more heavily congested network of Fig. 16

ρ	$P_b(\Omega)$	$P_b(\Omega_X)$ {%gain} [†]	$P_b(\Omega^*)$ {%gain} [*]	total gain ^{**}	Ω_X	Ω^*
3.5	0.393439	0.392404 {0.2631}	0.392404 {0.0}	0.2631	{5,6,3,3,6}	{5,6,3,3,6, 8,8,8,8,8,8,6}
4.5	0.498875	0.495574 {0.6617}	0.495564 {0.0020}	0.6637	{5,6,3,3,6}	{5,6,3,3,6, 8,8,7,8,8,8,6}
5.5	0.574471	0.568364 {1.0631}	0.568352 {0.0021}	1.0652	{4,6,2,2,6}	{4,6,2,2,6, 6,6,7,8,8,8,4}
6.0	0.604512	0.597356 {1.1837}	0.597341 {0.0025}	1.1862	{4,6,2,2,6}	{4,6,2,2,6, 6,6,7,8,8,8,4}
6.5	0.630650	0.622929 {1.2243}	0.622913 {0.0026}	1.2269	{4,6,2,2,6}	{4,6,2,2,6, 6,6,7,8,8,8,4}
7.0	0.653579	0.645607 {1.2198}	0.645590 {0.0025}	1.2223	{4,6,2,2,6}	{4,6,2,2,6, 6,6,7,8,8,8,4}
8.0	0.691874	0.682415 {1.3671}	0.682415 {0.0}	1.3671	{2,6,1,1,6}	{2,6,1,1,6, 3,3,8,7,7,7,2}
9.0	0.722550	0.710954 {1.6050}	0.710954 {0.0}	1.6050	{2,6,1,1,6}	{2,6,1,1,6, 3,3,8,7,7,7,2}
10	0.747659	0.735114 {1.6779}	0.735104 {0.0013}	1.6792	{2,6,1,1,6}	{2,6,1,1,6, 3,3,7,7,7,7,2}
20	0.867150	0.857811 {1.0770}	0.857787 {0.0029}	1.0799	{2,6,1,1,6}	{2,6,1,1,6, 3,3,7,7,7,7,2}

$$^{\dagger} \% \text{gain} = 100(P_b(\Omega) - P_b(\Omega_X)) / P_b(\Omega)$$

$$^* \% \text{gain} = 100(P_b(\Omega_X) - P_b(\Omega^*)) / P_b(\Omega_X)$$

$$^{**} \text{total gain} = 100(P_b(\Omega) - P_b(\Omega^*)) / P_b(\Omega)$$

4.2 Routing and Admission Control

To evaluate the relative benefits of routing and admission control, we investigated the performance that is achievable using different path sets. The routing scheme developed in [22, 23] was applied to the network of Fig. 10 to find path sets with minimum "congestion." The first step in this routing method is the generation of sets of highly node-disjoint paths between each of the source-destination (SD) pairs by means of a modified version of Dijkstra's shortest-path algorithm. In our case, the five SD pairs consist of the end nodes of each of the five circuits; we call the pair of end nodes of circuit 1 "SD pair 1," etc. A total of 11 paths were found—three between each of SD pairs 1, 2, and 3, and one between SD pairs 4 and 5.

Table 5 — Set of paths between the 5 SD pairs of Fig. 10
found by the modified Dijkstra's algorithm

SD Pair	Path Number	Nodes Traversed						
1	0	1	5	7	9			
	1	1	2	4	7	9		
	2	1	3	6	8	7	9	
2	3	2	1	3				
	4	2	5	3				
	5	2	4	7	8	6	3	
3	6	4	5	6				
	7	4	7	8	6			
	8	4	2	1	3	6		
4	9	8	7	10				
5	10	5	7					

This set of paths was then used as input to a program that performs an exhaustive search—of all possible sets of exactly one path between each SD pair—to find the path set(s) that minimizes congestion¹⁹ within the network. Two path sets were found to have the minimum congestion under the metric used. One is the set of paths used in Fig. 10, which we now denote as path set {0,3,6,9,10}, or the “nominal path set.” Here the path numbers refer to those shown in Table 5. The second path set {1,3,6,9,10}, which also has minimum congestion, is shown in Fig. 17. We refer to this as the “improved path set” for reasons that will soon become apparent. The most heavily congested path set {2,5,7,9,10}, which we refer to as the “worst path set,” is shown in Fig. 18.

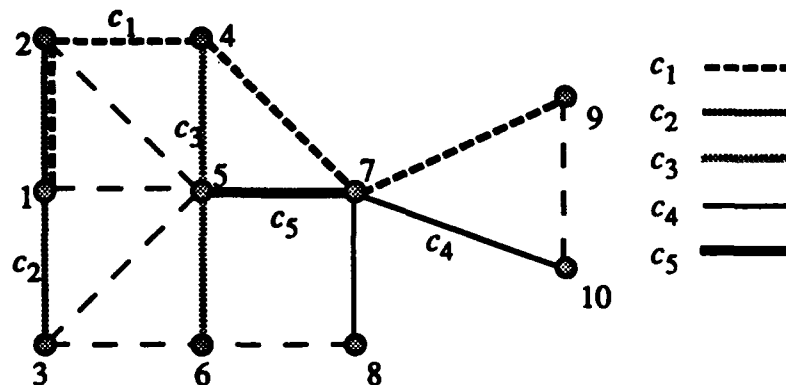


Fig. 17 — The improved path set {1,3,6,9,10}

¹⁹ This program computes congestion by taking a weighted sum of the number of shared nodes between all pairs of paths. In [22] we describe several different measures of congestion that were evaluated in our study of neural network models; according to the congestion measure used here, the worst path set is three times more congested than the improved path set.

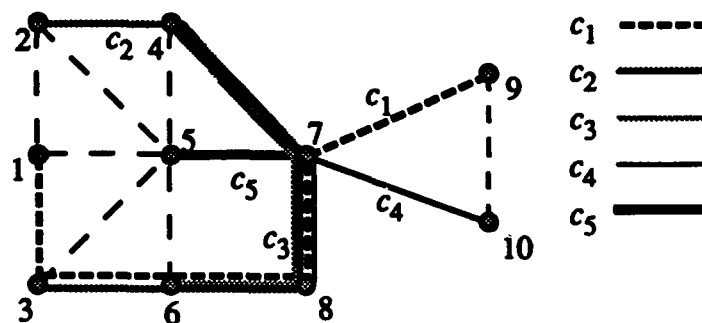


Fig. 18 — The worst path set {2,5,7,9,10}

Our studies have shown that the degree of improvement (if any) achieved through admission control depends on the path sets that are used. For example, an examination of Fig. 18 reveals that all five paths intersect at node 7. Thus, the five circuits are competing for a single (set of) resource, which results in the higher blocking probability and the lower throughput that is shown in Fig. 19. Because of the one-to-one relationship between the number of calls in the network and the number of transceivers in use at node 7, the results of the analysis agree with intuition for this network: Blocking any incoming call to reserve resource for a call on different circuit causes an increase in the blocking probability, even if the expected call has a much higher arrival rate than the blocked call. The best control policy for this network is no control.

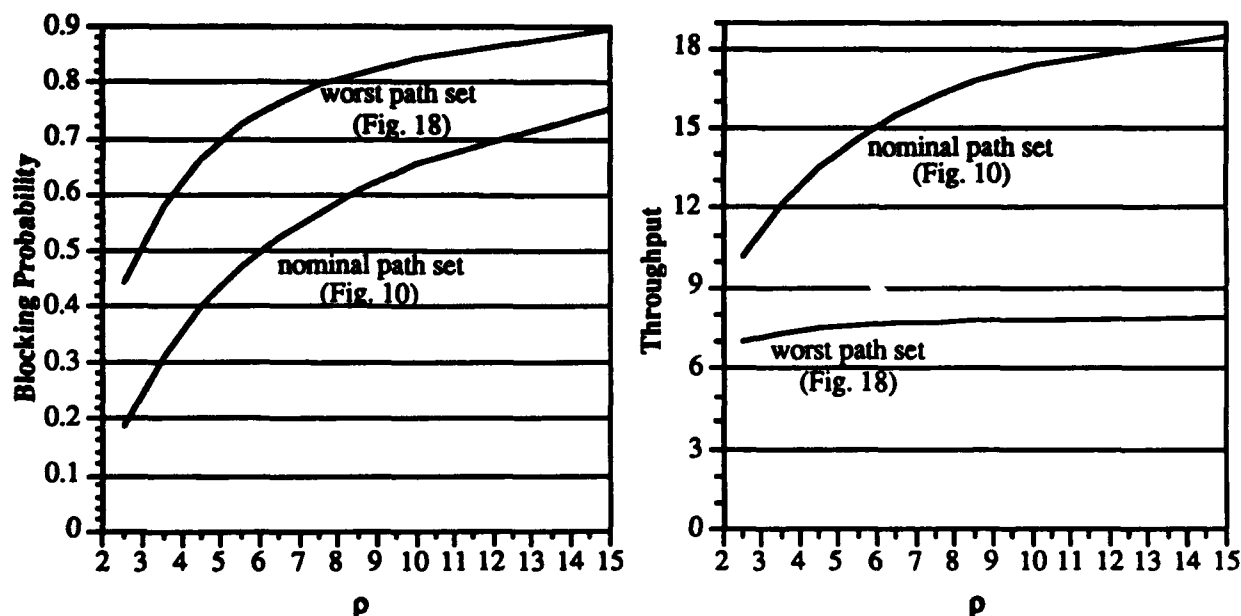


Fig. 19 — A comparison of the nominal path set and the worst path set in terms of blocking probability and throughput (in the worst path set, the optimal policy is to administer no control, i.e., $\Omega^* = \{6,6,6,6,6,8,\dots,8\}$)

The more interesting question concerns the effect of routing and control on the two minimum-congestion networks. By merely comparing the system and control constraints for

these networks, which are shown below, one is led to suspect that the configuration of Fig. 17 may provide better performance than that of Fig. 10. One indication of this is the fact that the nominal path set has two nodes (nodes 5 and 7) with the maximum nodal degree of three (here, we define nodal degree to be the number of circuits of which a given node is an element), but the improved path set has only one node (node 7) with the maximum nodal degree. Although the routing congestion for the two path sets is identical, the system and control constraints of both systems seem to indicate that the network of Fig. 17 is less constrained.

SYSTEM CONSTRAINTS

Improved Path Set	Nominal Path Set
$x_j \leq \min_{i: \text{node } i \in \text{circuit } j} \{T_i\},$	$x_j \leq \min_{i: \text{node } i \in \text{circuit } j} \{T_i\},$
$x_1 + x_2 \leq \min\{T_1, T_2\}$	$x_1 + x_2 \leq T_1$
$x_1 + x_3 \leq T_4$	$x_1 + x_3 + x_5 \leq T_5$
$x_3 + x_5 \leq T_5$	$x_1 + x_4 + x_5 \leq T_7$
$x_1 + x_4 + x_5 \leq T_7$	

CONTROL CONSTRAINTS

Improved Path Set	Nominal Path Set
$x_j \leq X_j$	$x_j \leq X_j$
$x_1 + x_4 \leq Y'_1$	$x_1 + x_3 \leq Y_1$
$x_1 + x_5 \leq Y'_2$	$x_1 + x_4 \leq Y_2$
$x_4 + x_5 \leq Y'_3.$	$x_1 + x_5 \leq Y_3$
	$x_3 + x_5 \leq Y_4$
	$x_4 + x_5 \leq Y_5$

In Figs. 20 and 21, curves of the network performance versus ρ for the nominal and the improved path sets show the effects of routing in our example. The data supporting these figures, and the optimal policies as ρ increases, are presented in Table A5. The figures also show the effects of admission control by presenting curves of the performance when control is used, and when it is not (uncontrolled), for both of the minimum congestion path sets. The performance measure in Fig. 20 is throughput, and in Fig. 21 it is blocking probability. The figures show that the improved path set (Fig. 17) does indeed provide better performance. At low ρ values (< 6) the uncontrolled performance of the improved path set is better than that of the controlled nominal path set (Fig. 10). At higher traffic rates, administering control on the nominal path set does yield better performance than is possible in an uncontrolled system that uses the improved path set; it also delivers a larger improvement percentage than does administering control on the improved path set. However, the use of an admission

control policy in conjunction with the improved path set clearly gives the best performance across the range of ρ values.

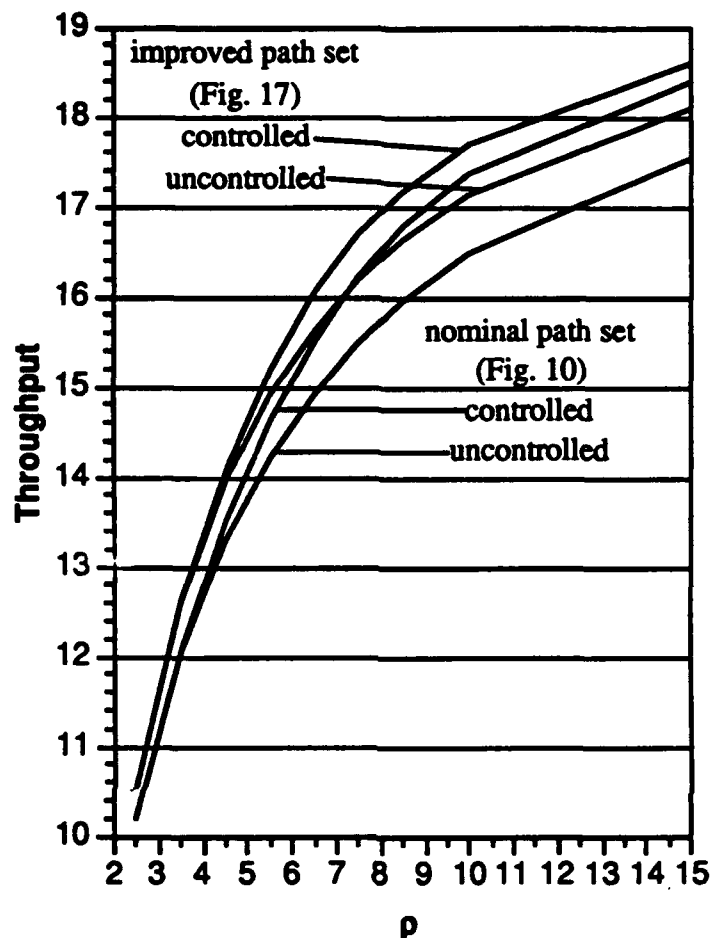


Fig. 20 — Throughput vs. loading (ρ) for the two minimum-congestion path sets

We can make two observations about the performance curves shown in Figs. 20 and 21. (1) The choice of path sets to reduce congestion can be an effective method for improving network performance. In fact, these figures indicate that it may be more effective than admission control. (2) However, the application of admission control does improve further the performance obtained by choice of good paths alone; at high traffic rates it produced approximately the same amount of improvement as the routing scheme.

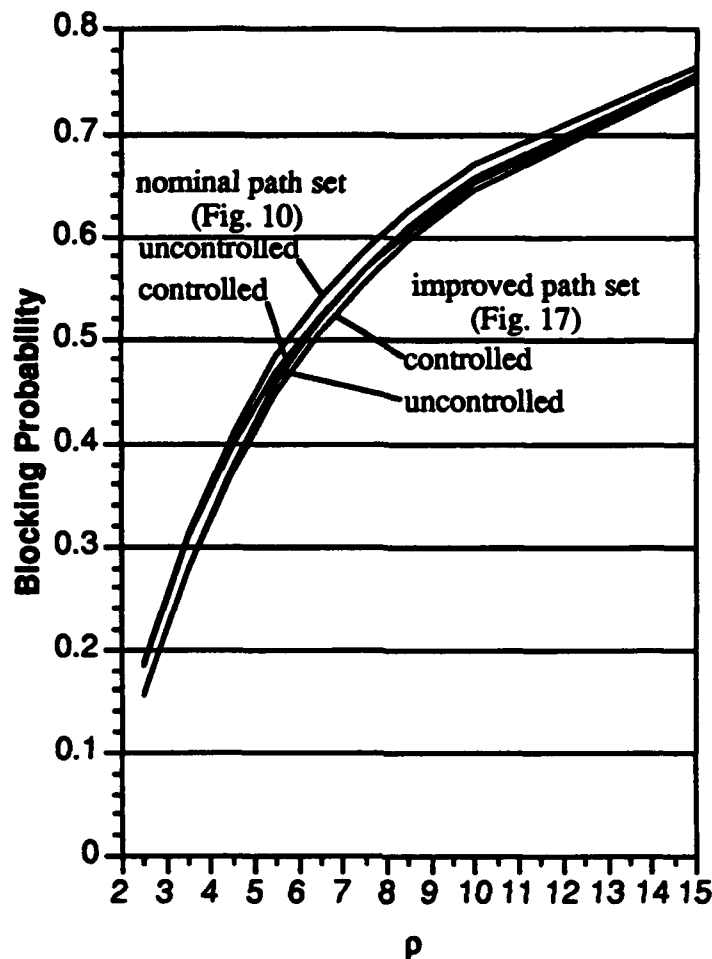


Fig. 21 — Blocking probability vs. loading (ρ) for the two minimum-congestion path sets

5 AN EXAMINATION OF SPECIAL NETWORK CLASSES

To gain further insight into the effects of voice admission control on network performance we have studied two different network classes. In Section 5.1 we examine the ubiquitous tandem network, which can be thought of as a building block of multihop networks. In Section 5.2 we study “multi-cross networks,” a class of networks that give insight into the reasons we have not been able to observe significant performance improvements through the use of admission control policies.

5.1 Analysis of Tandem Networks

One of the basic building blocks in multihop networks is the tandem subnetwork. A circuit-switched network may be viewed as a set of overlaid tandems between each source-destination pair. Clearly, there is a large degree of interaction among the tandems, as evidenced by the performance results we have presented thus far, thus making analysis and numerical evaluation difficult for all but relatively small networks. We have therefore decided

to investigate the case of an isolated tandem network to see if results could be obtained that would apply to more general network configurations.

5.1.1 A Five-Node Tandem

Consider the five-node tandem network shown in Fig. 22. We consider all circuits that consist of one- and two-hop calls; there are a total of seven in this network: four one-hop circuits (1, 2, 3, and 4) and three two-hop circuits (5, 6, and 7). At node 3, five circuits intersect (i.e., 2, 3, 5, 6, and 7); this is the largest number of circuits that can intersect in any length tandem network that allows only one- and two-hop calls.

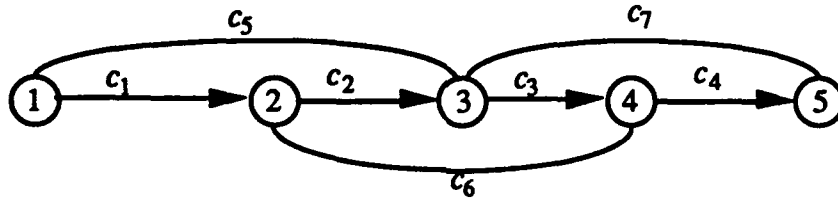


Fig. 22 — A five-node tandem network

The system constraints for this network are:

$$\begin{aligned} x_1 + x_5 &\leq T_1 \\ x_1 + x_2 + x_5 + x_6 &\leq T_2 \\ x_2 + x_3 + x_5 + x_6 + x_7 &\leq T_3 \\ x_3 + x_4 + x_6 + x_7 &\leq T_4 \\ x_4 + x_7 &\leq T_5 \end{aligned}$$

and the control constraints are:

Node 2:

$$\begin{aligned} x_1 + x_2 &\leq Y_1 & x_1 + x_2 + x_5 &\leq Y_7 \\ x_1 + x_5 &\leq Y_2 & x_1 + x_2 + x_6 &\leq Y_8 \\ x_1 + x_6 &\leq Y_3 & x_1 + x_5 + x_6 &\leq Y_9 \\ x_2 + x_5 &\leq Y_4 & x_2 + x_5 + x_6 &\leq Y_{10} \\ x_2 + x_6 &\leq Y_5 \\ x_5 + x_6 &\leq Y_6 \end{aligned}$$

Node 3:

$$\begin{aligned} x_2 + x_3 &\leq Y_{11} & x_2 + x_3 + x_5 &\leq Y_{18} & x_2 + x_3 + x_5 + x_6 &\leq Y_{27} \\ x_2 + x_7 &\leq Y_{12} & x_2 + x_3 + x_6 &\leq Y_{19} & x_2 + x_3 + x_5 + x_7 &\leq Y_{28} \\ x_3 + x_5 &\leq Y_{13} & x_2 + x_3 + x_7 &\leq Y_{20} & x_2 + x_3 + x_6 + x_7 &\leq Y_{29} \\ x_3 + x_6 &\leq Y_{14} & x_2 + x_5 + x_7 &\leq Y_{21} & x_2 + x_5 + x_6 + x_7 &\leq Y_{30} \\ x_3 + x_7 &\leq Y_{15} & x_2 + x_6 + x_7 &\leq Y_{22} & x_3 + x_5 + x_6 + x_7 &\leq Y_{31} \\ x_5 + x_7 &\leq Y_{16} & x_3 + x_5 + x_6 &\leq Y_{23} \\ x_6 + x_7 &\leq Y_{17} & x_3 + x_5 + x_7 &\leq Y_{24} \\ & & x_3 + x_6 + x_7 &\leq Y_{25} \\ & & x_5 + x_6 + x_7 &\leq Y_{26} \end{aligned}$$

Node 4:

$$\begin{aligned} x_3 + x_4 &\leq Y_{32} & x_3 + x_4 + x_6 &\leq Y_{35} \end{aligned}$$

$$\begin{array}{ll} x_4 + x_6 \leq Y_{33} & x_3 + x_4 + x_7 \leq Y_{36} \\ x_4 + x_7 \leq Y_{34} & x_4 + x_6 + x_7 \leq Y_{37} \end{array}$$

Although the dimensionality of the state space is equal to the number of circuits (7 in this example), the number of degrees of freedom (i.e., the number of parameters that need to be optimized), and hence the dimensionality of the optimization problem, is equal to the number of linear inequality constraints (thresholds and control constraints) that must be considered when specifying the control policy. Thus, the dimensionality of the state space is not the sole contributor to the dimensionality of the optimization problem. The complexity of the problem in this seemingly innocent tandem network virtually explodes as a result of the high degree of circuit interleaving (not the number of circuits). A descent search of this network (using parameters $T_i = 8$ for all nodes i ; $X_j \leq 6$ and $\rho_j = 5$ for all circuits j) was programmed in C; it ran for nearly 29 days (28 days, 20 hours, 28 minutes) on an IBM RISC 6000 without converging. After 7 iterations (2715 policies) of the threshold-only loop of the search program, the best threshold-only solution $\{6,5,5,6,5,0,5\}$ was found to have $P_b = 0.613793$, which is a 2.419437% improvement over the uncontrolled solution ($P_b = 0.629012$). The best solution we have found has $P_b = 0.613519$ for the control policy

$$\Omega^* = \{6,5,5,6,5,0,5, \quad \leftarrow \text{thresholds} \\ 8,8,6,6,5,5,5,8,8,8,8,8,5,5,8,5,8,8,8,8,8,8,8,8,8,8,8,6,8,8,8,8\}, \leftarrow \text{constraints}$$

which represents a 2.463% reduction in blocking probability from the uncontrolled system.

The implementation of the descent search that we used for this problem was reasonable for the threshold-only portion of the search, but not for the search of the entire space. The dimensionality of the threshold-only search problem, and hence the number of control policies that must be evaluated to find a descent direction, is easily handled. However, in the combined portion of the search where we were attempting to optimize both the thresholds (the X_j 's) and the control constraint limits (the Y_I 's), our search procedure was overly ambitious. The time required to evaluate a single point, and the increased dimensionality, which causes a large increase in the number of policies that must be evaluated to determine a descent direction, combined to make our search impractical. We estimate that the search would need another 1058 days (approximately three years) to converge. Clearly, practicality considerations require the use of a more efficient, but less thorough search. Let us explain with an example.

We will start the example at nearly the end of the threshold-only loop of the descent search. We have just moved to the point $\Omega_0 = \{6,5,5,6,5,0,5\}$, i.e., $X_1 = 6, X_2 = 5, \dots, X_7 = 6$. The Y_I are not listed because they are all fixed at 8, their maximum value. Now, the immediate neighborhood of Ω_0 must be searched for policies that give better performance. For the threshold portion of the search, we define the immediate neighborhood to be all

admissible policies Ω_n (policies that do not violate the system constraints) that can be obtained by changing at most three of the threshold values in Ω_0 by ± 1 . We systematically select sets of three parameters to be varied, eventually selecting all the $\binom{7}{3} = 35$ different parameter triplets. For each triplet (X_i, X_j, X_k) , we evaluate all combinations of parameter values within ± 1 of the Ω_0 value. Each member of the selected triplet can take either two or three different values. If its value in the Ω_0 policy is equal to a system constraint (e.g., $X_1 = 6$ or $X_6 = 0$) it can only take one of two values in the Ω_n policy (e.g., $X_1 = 5$ or 6 , $X_6 = 0$ or 1). Triplet members whose values in the Ω_0 policy are not equal to a system constraint (e.g., $X_2 = 5$) can take one of three values in the Ω_n policy (e.g., $X_2 = 4, 5$, or 6). For example, when $i = 1, j = 2$, and $k = 6$, the following policies must be evaluated:

$$\begin{aligned} \Omega_1 &= \{5, 4, 5, 6, 5, 0, 5\}, & \Omega_2 &= \{5, 4, 5, 6, 5, 1, 5\}, & \Omega_3 &= \{5, 5, 5, 6, 5, 0, 5\}, \\ \Omega_4 &= \{5, 5, 5, 6, 5, 1, 5\}, & \Omega_5 &= \{5, 6, 5, 6, 5, 0, 5\}, & \Omega_6 &= \{5, 6, 5, 6, 5, 1, 5\}, \\ \Omega_7 &= \{6, 4, 5, 6, 5, 0, 5\}, & \Omega_8 &= \{6, 4, 5, 6, 5, 1, 5\}, & \Omega_9 &= \{6, 5, 5, 6, 5, 0, 5\}, \\ \Omega_{10} &= \{6, 5, 5, 6, 5, 1, 5\}, & \Omega_{11} &= \{6, 6, 5, 6, 5, 0, 5\}, & \Omega_{12} &= \{6, 6, 5, 6, 5, 1, 5\}. \end{aligned}$$

It can be seen that, for each triplet of parameter values, a minimum of $2^3 = 8$ policies to a maximum of $3^3 = 27$ policies must be evaluated. Since there are 35 parameter triplets, somewhere between 280 and 945 policies must be evaluated to search the immediate neighborhood. A search of the immediate neighborhood of Ω_0 found no policies that give better performance. Therefore we conclude that Ω_0 is the optimal threshold-only policy.

Before proceeding to the combined loop of the search, let us discuss the rationale for simultaneously varying multiple parameters. We have found that varying each of the thresholds individually may not allow the search to converge to the optimal policy. However, empirical studies indicate that when several parameters are varied simultaneously, the search will converge to the optimal policy from any starting point. How many are necessary is uncertain; in the limit all parameters must be varied simultaneously, and the search approaches an exhaustive one. It appears, however, that such extreme measures are not necessary. In our smaller examples we have not seen any improvement by increasing the number of simultaneously varied parameters beyond two or three.

In the combined portion of the search, the need to simultaneously vary multiple parameters can be more readily seen. The benefit of raising the value of some Y_l cannot be measured if the X_j , ($j \in I$) cannot also be simultaneously increased. In the present tandem network, because Y_{27} through Y_{31} each influence four different thresholds, it appears that varying five parameters simultaneously may be necessary.

Returning to our descent search example, we start the combined loop from the state Ω_0 , which now includes Y_I . In this loop, we search the immediate neighborhood until we find a policy Ω' with improved performance. We immediately jump to this new point ($\Omega_0 = \Omega'$) and restart the combined search. The immediate neighborhood is larger in this portion of the search. It includes all feasible policies Ω_n that can be obtained by changing at most five of the parameter values in Ω_0 by ± 1 . The search proceeds in a manner similar to that used in the threshold-only loop, except now we select sets of five parameters instead of triplets. This is where our search gets hung up. There are approximately 650,091 sets of five parameters that must be checked to evaluate all immediate neighbors. For each of these sets somewhere between $2^5 = 32$ and $3^5 = 243$ different policies must be evaluated. Thus, a minimum of between 21×10^6 and 155×10^6 policies must be evaluated before concluding that the optimal policy has been found. Our program running on the IBM RISC computer can evaluate about 100 policies per minute, which gives the basis of our estimate of approximately three years to complete the search. Clearly, a different, more efficient approach is needed to make this search practical.

A Progressive-Search Method

We have examined a more efficient approach, which we call a "progressive" descent search. Instead of performing the search in two loops (threshold and combined) and varying five parameters at the outset of the combined loop, the progressive search starts by varying only one parameter at a time in a "1-parameter" loop. In the 1-parameter loop, the immediate neighborhood of Ω_0 is defined to be all admissible coordinate convex policies Ω' that can be obtained by changing one of the parameter values in Ω_0 by ± 1 . We systematically select and evaluate the system performance under different Ω' until a policy with improved performance is discovered, or until all the Ω' have been evaluated. If improved performance is possible using policy Ω' , the search "moves" to Ω' (i.e., we set $\Omega_0 = \Omega'$) and the procedure is repeated. In the search of the tandem network, this procedure was repeated 17 times. The best solution Ω^* (the same policy as was found in the original search) was found on the 16th iteration after evaluating a total of 193 policies; on the 17th iteration 50 more policies had to be evaluated to conclude no better policy could be found. If all the Ω' are evaluated without finding one that yields better performance, the search progresses to the 2-parameter loop.

The 2-parameter search starts from the point Ω_0 that was found in the 1-parameter loop. It proceeds in the same manner as the 1-parameter search, except now the immediate neighborhood of Ω_0 is defined to be all feasible coordinate convex policies Ω' that can be obtained by changing at most two of the parameter values in Ω_0 by ± 1 . In the search of

policies for the tandem network, 2,379 different policies are in the 2-parameter immediate neighborhood, but none deliver better performance than Ω^* . The evaluation of these policies took about 54 minutes.

The search has completed the 3-parameter (72,739 policies) and 4-parameter (1,615,987 policies) loops without finding a better policy than Ω^* . It was scheduled to perform a 5-parameter loop as well. The 5-parameter search plan was never completed because of the large number of policies that had to be evaluated. To perform the 5-parameter loop another $(\binom{44}{5} \times 2^4 >) 17 \times 10^6$ to $(\binom{44}{5} \times 2 \times 3^4 >) 175 \times 10^6$ policies must be evaluated to verify that Ω^* is the best policy. If a better policy is found in this loop, the number of evaluations may be much larger. At a rate of 100 policies per minute (which is about the average), it will take between four months and 3.5 years to complete these two loops. Therefore, realistically the search had to be terminated at some point before the 5-parameter search is completed.

This example illustrates the complexity of the problem. Although our descent search methods are quite effective in reducing the number of policies that must be evaluated, it is not practical to expect to complete the search, even in some apparently simple networks. Nonetheless, these search methods do serve as heuristic techniques that can provide a good policy in a reasonable amount of time.

5.1.2 A Four-Node Tandem

Reducing the network to a 4-node tandem as shown in Fig. 23 significantly reduces the dimensionality of the problem. Again, we only consider one- and two-hop calls. The maximum number of circuits intersecting at a node is 4, which occurs at nodes 2 and 3.

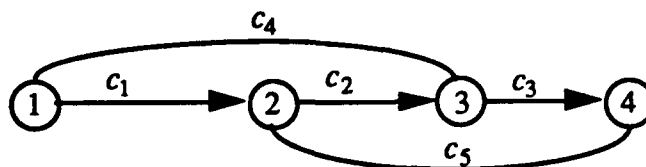


Fig. 23 — A four-node tandem network

The system constraints for this network are:

$$\begin{aligned} x_1 + x_4 &\leq T_1 \\ x_1 + x_2 + x_4 + x_5 &\leq T_2 \\ x_2 + x_3 + x_4 + x_5 &\leq T_3 \\ x_3 + x_5 &\leq T_4 \end{aligned}$$

and the control constraints are:

Node 2:

$$\begin{aligned} x_1 + x_2 &\leq Y_1 & x_1 + x_2 + x_4 &\leq Y_7 \\ x_1 + x_4 &\leq Y_2 & x_1 + x_2 + x_5 &\leq Y_8 \end{aligned}$$

$$\begin{array}{ll}
x_1 + x_5 \leq Y_3 & x_1 + x_4 + x_5 \leq Y_9 \\
x_2 + x_4 \leq Y_4 & x_2 + x_4 + x_5 \leq Y_{10} \\
x_2 + x_5 \leq Y_5 & \\
x_4 + x_5 \leq Y_6 &
\end{array}$$

Node 3:

$$\begin{array}{ll}
x_2 + x_3 \leq Y_{11} & x_2 + x_3 + x_4 \leq Y_{14} \\
x_3 + x_4 \leq Y_{12} & x_2 + x_3 + x_5 \leq Y_{15} \\
x_3 + x_5 \leq Y_{13} & x_3 + x_4 + x_5 \leq Y_{16}
\end{array}$$

Thus, by eliminating one node in the tandem network, the search space is reduced from seven call types and 37 constraints to five call types and 16 constraints. A descent search for the optimal control policy (using the same parameters as were used for the five-node tandem) has been completed. The solution following the threshold only portion of the search has $P_b = 0.607774$ for the policy $\{6,3,6,3,3\}$, which is a 0.3676% improvement over the uncontrolled system. The optimal policy is $\{6,3,6,3,3,8,8,8,3,3,8,8,8,3,8,8,8,8\}$ and has $P_b = 0.596042$.

5.2 Multi-Cross Networks and Network Self Regulation

In an attempt to determine the reason that such a small degree of performance improvement was achieved through the use of admission control, we have examined the network configuration of Fig. 24, which we refer to as a "multi-cross network." The horizontal circuit c_0 intersects each of the remaining N circuits, c_1, \dots, c_N . Circuits c_1 through c_N are mutually disjoint and only share resources with circuit c_0 . The network of Fig. 10, which has served as the primary testing ground for our admission-control studies, is similar to the multi-cross network since circuits c_2, c_3 , and c_4 each intersect circuit c_1 at one node, and do not intersect at all with each other. However, it differs in that circuit c_5 intersects with circuit c_1 at two nodes, and with circuits c_3 , and c_4 at one node each. Nevertheless, it is reasonable to expect that an examination of multi-cross networks should be able to provide some degree of insight into the operation of more complex networks.

Intuitively, it can be seen that at high utilization rates or at high N (the number of mutually disjoint circuits intersecting circuit c_0) optimal admission control will restrict the number of calls allowed on circuit c_0 , ultimately not admitting any calls on this circuit. Intuition might also suggest that the performance gain obtained by administering such control would continue to grow with N .

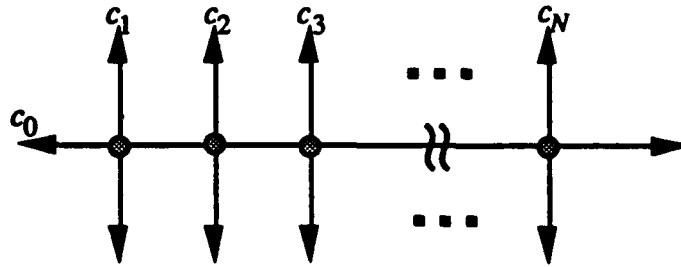


Fig. 24 — An example of a multi-cross network

We have examined these hypotheses by applying our descent search method to a series of multi-cross networks with increasing N and fixed utilization rates and capacities. In these studies, we have set the node capacity $T_i = \text{circuit capacity} = 6$, $\rho_0 = 14$, and ρ_1 through $\rho_N = 7$. Since at most only two circuits intersect at any node, there are no control constraints to adjust. Thus, the descent search attempts to find the optimum value of the $N + 1$ circuit thresholds. The results of these efforts for $N = 1, \dots, 9$ are shown in terms of blocking probability in Figs. 25 and 26, and in terms of performance gain in Fig. 27. As our analysis of Section 4 revealed, when equally weighted services are competing for a single (set of) resource (e.g., in a multi-cross network with $N = 1$) the best strategy is to administer no control—all calls are accepted as long as resources are available to serve them. Thus, in Figs. 25-27 the optimal policy and the uncontrolled policy are the same when $N = 1$. However, at the high utilization rates ($\rho_0 = 14$, $\rho_1 - \rho_N = 7$) assumed in this example, the optimal policy for $N > 1$ is $\Omega^* = \{0, 6, 6, \dots\}$.

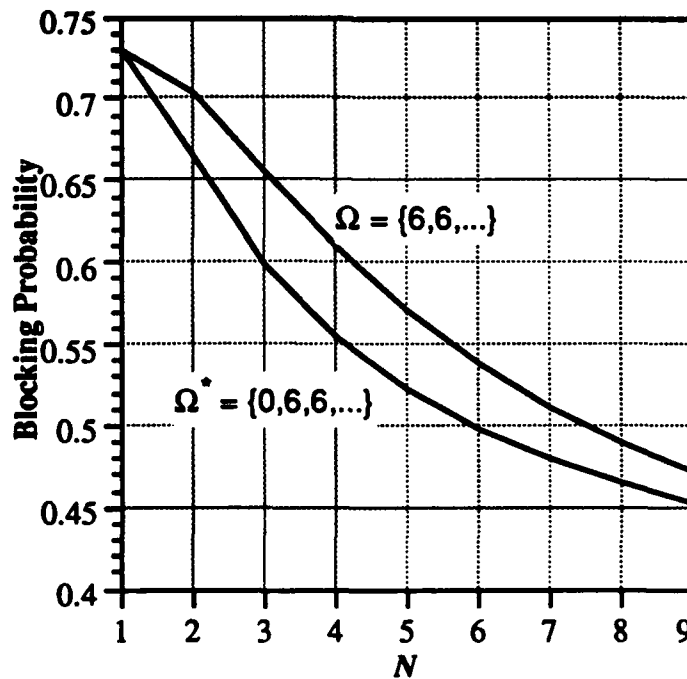


Fig. 25 — Blocking probability in multi-cross network vs. N

It is noteworthy that, as shown in Fig. 25, blocking probability decreases as N increases. In a multi-cross network with equal utilization rates on the N crossing circuits ($c_1 - c_N$),

$$P_b(\Omega) = \frac{\rho_0 P_{b0}(\Omega) + N \rho_1 P_{b1}(\Omega)}{\rho_0 + N \rho_1},$$

since the circuit blocking probability $P_{bj}(\Omega) = P_{b1}(\Omega)$ for $j \geq 1$, i.e., the circuit blocking probability is the same for all the N mutually disjoint crossing circuits. In the optimally controlled network with $N > 1$ and $\Omega = \Omega^* = \{0, 6, 6, \dots\}$, $P_{b0}(\Omega^*) = 1.0$ and $P_{b1}(\Omega^*) = 0.33133$ (with $\rho_1 = 7$, $T_i = 6$). Thus, it is easy to see that

$$\lim_{N \rightarrow \infty} P_b(\Omega^*) = P_{b1}(\Omega^*).$$

This limit holds in the uncontrolled network as well. With $\Omega = \{6, 6, \dots\}$, $P_{b0}(\Omega) < 1.0$ and $P_{b1}(\Omega) > P_{b1}(\Omega^*)$ for $1 < N < \infty$. As shown in Fig. 26, $P_{b0}(\Omega)$ increases with N ; as it approaches 1.0, $P_{b1}(\Omega)$ approaches $P_{b1}(\Omega^*)$ and the uncontrolled overall blocking probability $P_b(\Omega)$ approaches $P_b(\Omega^*)$, the value obtained by applying the optimal policy. Furthermore, the limit holds for all values of ρ_0 and ρ_1 greater than zero. Thus, the arbitrary loading used in the present example gives representative results for any loading ($\rho_1, \dots, \rho_N > 0$).

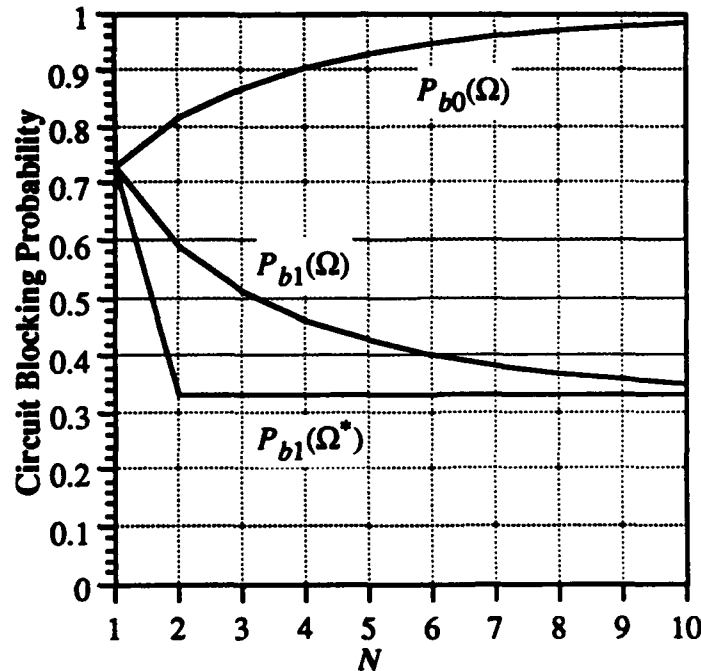


Fig. 26 — Circuit blocking probability vs. N

These observations give an explanation for the relatively small performance improvements shown in Fig. 27, and seen in our studies of Section 4. They show that circuit-switched networks tend to be self regulating. In the case of a multi-cross network, as N

increases there is increased likelihood that the resources needed to complete a call on circuit c_0 will be seized by one of the N multi-cross circuits. Thus, without administering any control, calls on c_0 will have reduced access to the network and calls on the crossing circuits ($c_1 - c_N$) will see improved access. In a general network setting with equally weighted calls, this self-regulation phenomenon will tend to admit more calls on the circuits that compete with the fewest other circuits, and reject more calls on the circuits that must compete with many other circuits for resources. Our studies have shown that the best form of admission control is to restrict the access of those circuits that interfere with many other circuits. Thus the application of admission control and the result of self regulation achieve nearly the same network performance. Network performance generally can be improved by administering control, but the gain relative to the uncontrolled system is limited by the network's natural self regulation.

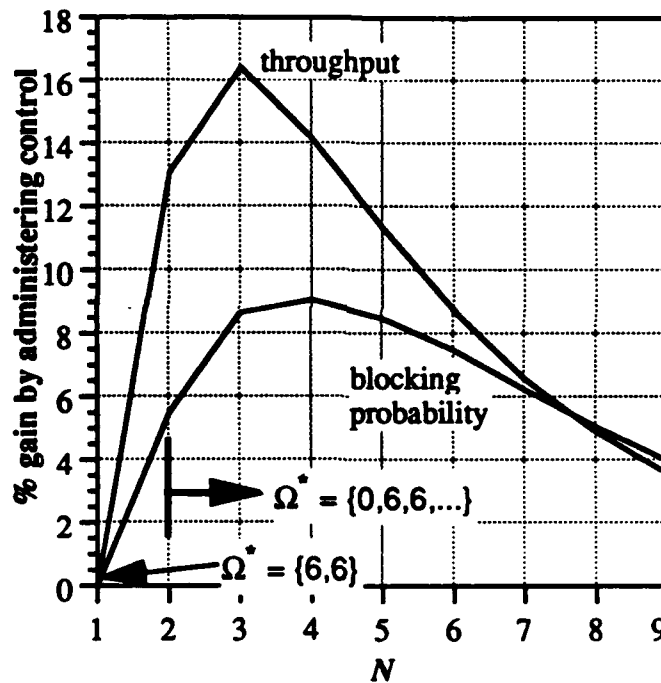


Fig. 27 — Throughput and blocking probability gains vs. N

6 EXTENSION TO INTEGRATED NETWORKS

Thus far in our presentation, we have focused primarily on the voice-call process and the corresponding performance measures of blocking probability and throughput. A necessary part of calculating these performance metrics is the tedious calculation of the normalization constant $\pi(0)$. Once the normalization constant is known, however, it is a simple step to obtain the steady state distribution $\pi(x)$ of the call process. This information can then be used to estimate the performance of an integrated voice/data network not only in terms of the

voice metrics, but also in terms of certain data metrics and weighted sums of the voice and data metrics.

From the steady state distribution of the voice call process, we can calculate the expected voice state $\bar{x} = \{\bar{x}_j, j = 1, \dots, J\}$ as follows:

$$\bar{x}_j = \sum_{n=1}^{x_j} n P(x_j = n) = \sum_{n=1}^{x_j} \left\{ n \sum_{\{x: x_j = n\}} \pi(x) \right\}.$$

Assuming that data traffic is free to use a resource (transceiver) whenever it is not occupied by voice traffic, the "residual data capacity" of an integrated network with given voice utilization rates is an indicator of potential network performance with respect to data. We have considered two different measures of residual data capacity: *mean data capacity*, denoted C_m , and *bottleneck data capacity*, denoted C_b .

Both residual data capacity metrics require a node-based evaluation of the network. We define the data capacity of node i , which we denote as C_i , to be its expected data transmission capacity (i.e., the average number of transceivers not involved in voice calls). Thus

$$C_i = T_i - \sum_{j=1}^J c_{ji} \bar{x}_j,$$

where T_i is the number of transceivers at node i , and

$$c_{ji} = \begin{cases} 1, & \text{node } i \in \text{circuit } j \\ 0, & \text{otherwise} \end{cases}$$

as defined previously. Now we can write the mean data capacity as

$$C_m = \frac{1}{N} \sum_{i=1}^N C_i,$$

and the bottleneck data capacity as

$$C_b = \min_{i=1}^N \{C_i\},$$

where N = the number of nodes in the network.

These metrics require no knowledge of the data traffic statistics, nor do they exploit such knowledge (other than through the choice of C_m or C_b). They give a gross estimate of the network's ability to handle arbitrary data loading in the presence of a given voice traffic load. Because these data metrics can be relatively easily calculated once the voice

distribution is known, they can quite naturally be used in a metric Θ that consists of the weighted sum of the residual data capacity and voice throughput.

$$\Theta(\Omega) = \beta C_* + \Gamma(\Omega).$$

Here β is the weighting factor, which is chosen to reflect the relative importance of the voice and data performance metrics, and C_* is used to denote C_m or C_b . Furthermore, Θ can be maximized by using our descent-search method (appropriately converted to an ascent search), thus giving a good control policy for the integrated network.

In the future we plan to extend these results to incorporate performance measures for data traffic that provide a more meaningful characterization, such as delay. In addition we plan to investigate network control schemes in which admission-control policies for voice traffic are permitted to depend on data-packet queue sizes. However, such an extension would require a major change in the mathematical formulation because the voice traffic process would no longer be characterized by a product-form solution. Finally, we intend to study schemes in which the instantaneous value of the residual capacity is utilized.

7 CONCLUSIONS

We have addressed the issue of voice-call admission control in wireless networks. We have used the MSMR methodology of Jordan and Varaiya, and introduced a modified search technique for performance optimization. We have demonstrated that there exists a natural tendency for networks to be self-regulating, and, consequently, the exercise of admission control achieves minimal improvement in blocking probability or throughput performance as long as all call types are weighted equally. However, we have shown that admission control is imperative in cases of unequally weighted call types. Furthermore, we have shown the intimate and sensitive connection between admission control and routing, the latter being also a significant factor affecting network performance. In all cases we have found that optimizing only the circuit thresholds (in which case the problem dimensionality is equal to the state space dimensionality) results in control policies that are almost as good as the optimal policy. Thus, good policies can be obtained with a much simpler search than is required to optimize both the thresholds and the control constraints.

We have also shown how the analysis presented here can be applied to an integrated voice-data environment in which voice can be separated from data operation by means of boundary-based models, such as the one presented in [14]. We have exploited the product-form circuit-switched voice model to derive data related metrics in an integrated voice-data network. However, we have not yet completed the study of such an extension.

REFERENCES

1. D. J. Baker, A. Ephremides, and J. A. Flynn, "The Design and Simulation of a Mobile Radio Network with Distributed Control," *IEEE Journal on Selected Areas in Communications*, SAC-2 pp. 226 - 237, 1984.
2. A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proceedings of the IEEE*, 75-No. 1 pp. 56 - 73, 1987.
3. J. E. Wieselthier and A. Ephremides, "Voice/Data Integration in Mobile Radio Networks: Overview and Future Research Directions," NRL Report 9189, Naval Research Laboratory, September 1989.
4. F. P. Kelly, "Blocking Probabilities in Large Circuit-Switched Networks," *Adv. Appl. Prob.*, 18 pp. 473-505, 1986.
5. K. W. Ross and D. Tsang, "Teletraffic Engineering for Product-Form Circuit-Switched Networks," *Adv. Appl. Prob.*, 22 pp. 657-675, 1990.
6. I. Viniotis and A. Ephremides, "Optimal Switching of Voice and Data at a Network Node," *Proceedings of the 26th Conference on Decision and Control*, pp. 1504-1507, December 1987.
7. I. Viniotis and A. Ephremides, "On the Optimal Dynamic Switching of Voice and Data in Communication Networks," *Computer Networking Symposium*, pp. 8-16, 1988.
8. J. E. Wieselthier and A. Ephremides, "A Study of Channel-Access Schemes for Integrated Voice/Data Radio Networks," NRL Report 9359, Naval Research Laboratory, November 29 1991.
9. S. Jordan and P. Varaiya, "Control of Multiple Service, Multiple Resource Communication Networks," *Proceedings of IEEE INFOCOM'91*, pp. 648 - 657, April 1991.
10. S. Jordan and P. Varaiya, "Throughput in Multiple Service, Multiple Resource Communication Networks," *IEEE Transactions on Communications*, 39-No. 8 pp. 1216 - 1222, 1991.
11. E. Geraniotis and I.-H. Lin, "Control of Integrated Voice/Data Multihop Radio Networks Via Reduced-Load Approximations," to appear as an NRL Memorandum Report, Naval Research Laboratory, 1993.
12. K. W. Ross and D. H. K. Tsang, "The Stochastic Knapsack Problem," *IEEE Transactions on Communications*, 37-No. 7 pp. 740 - 747, 1989.
13. S.-P. Chung and K. W. Ross, "Reduced Load Approximations for Multirate Loss Networks," preprint, 1991.
14. J. E. Wieselthier and A. Ephremides, "A Movable-Boundary Channel-Access Scheme for Integrated Voice/Data Networks," *Proceedings of IEEE INFOCOM'91*, pp. 721-731, 1991.

15. J. M. Aein, "A Multi-User-Class, Blocked-Calls-Cleared, Demand Access Model," *IEEE Transactions on Communications*, COM-26-No. 3 pp. 378 - 385, 1978.
16. G. J. Foschini and B. Gopinath, "Sharing Memory Optimally," *IEEE Transactions on Communications*, COM-32-No. 3 pp. 352-360, 1983.
17. S. Jordan and P. Varaiya, "A Continuous Model of Multiple Service, Multiple Resource Communication Networks Part 1: Theory," submitted to *IEEE Transactions on Communications*, 1991.
18. S. Jordan and P. Varaiya, "A Continuous Model of Multiple Service, Multiple Resource Communication Networks Part 2: Algorithms," submitted to *IEEE Transactions on Communications*, 1991.
19. A. E. Conway and N. D. Georganas, *Queueing Networks—Exact Computational Algorithms: A Unified Theory Based on Decomposition and Aggregation*, Cambridge, Mass.: The MIT Press, 1989.
20. D. Bertsekas and R. Gallager, *Data Networks*, pp. 382-403, Englewood Cliffs: Prentice Hall, 1987.
21. Wolfram Research, Inc., *Mathematica*, Version 2.0, Wolfram Research, Inc.: Champaign, Illinois, 1991.
22. J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "A Neural Network Approach to Routing in Multihop Radio Networks," *Proceedings of IEEE INFOCOM'91*, pp. 1074 - 1083, April 1991.
23. J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "A Neural Network Approach to Routing Without Interference in Multihop Radio Networks," *to appear in IEEE Transactions on Communications*, 1993.

APPENDIX

Table A1. — Blocking Probability and Throughput Under Optimal Policies
for the Network of Fig. 10 (see Section 4.1)

ρ	Circuit Thresholds					optimal P_b	uncont. P_b	%† gain	optimal thruput	uncont. thruput	%* gain
	1	2	3	4	5						
0.5	3	3	3	3	3	0.13528	0.13528	0.0	2.161905	2.161905	0.0
1.0	2	3	3	3	3	0.322029	0.322206	0.054934	3.389854	3.388969	0.026103
1.5	0	3	3	3	3	0.439654	0.446261	1.48052	4.202593	4.15304	1.193183
2.0	0	3	3	3	2	0.512198	0.529719	3.3076	4.878022	4.702812	3.725655
2.5	0	3	3	3	1	0.566943	0.589581	3.83968	5.413216	5.130241	5.515823
3.0	0	3	3	3	0	0.607692	0.634854	4.27846	5.884615	5.477187	7.438638
5	0	3	3	3	0	0.717797	0.743659	3.477683	7.055085	6.408533	10.08893
10	0	3	3	3	0	0.839239	0.850857	1.36545	8.038067	7.457160	7.789931

† % gain = $100 (P_b(\text{uncontrolled}) - P_b(\text{optimal})) / P_b(\text{uncontrolled})$

* % gain = $100 (\text{Thruput}(\text{optimal}) - \text{Thruput}(\text{uncontrolled})) / \text{Thruput}(\text{uncontrolled})$

Table A2 — Optimal Policies for the Network of Fig. 10
with two different ρ values (see Section 4.1)

ρ_1	ρ_{2+}	Circuit Thresholds					optimal P_b	uncont. P_b	%† gain	optimal thruput	uncont. thruput	%* gain
		1	2	3	4	5						
3	1	2	3	3	3	3	0.503803	0.505577	0.350904	3.473378	3.460959	0.358821
6	1	1	3	3	3	3	0.649511	0.654063	0.695910	3.504889	3.459372	1.315755
6	2	0	3	3	3	2	0.65157	0.679974	4.177282	4.878022	4.480360	8.875674
9	0.1	3	3	3	3	3	0.71592	0.71592	0.0	2.670357	2.670357	0.0
9	0.3	3	3	3	3	3	0.727096	0.727096	0.0	2.783617	2.783617	0.0
9	0.5	3	3	3	3	3	0.732078	0.732078	0.0	2.947146	2.947146	0.0
9	1	1	3	3	3	3	0.729024	0.735929	0.938362	3.522692	3.432918	2.615090
9	2	0	3	3	3	2	0.713058	0.743175	4.052545	4.878022	4.366025	11.72686
9	3	0	3	3	3	0	0.71978	0.755392	4.714285	5.884615	5.136778	14.55850

† % gain = $100 (P_b(\text{uncontrolled}) - P_b(\text{optimal})) / P_b(\text{uncontrolled})$

* % gain = $100 (\text{Thruput}(\text{optimal}) - \text{Thruput}(\text{uncontrolled})) / \text{Thruput}(\text{uncontrolled})$

$\rho_{2+} \Rightarrow \rho_2 = \rho_3 = \rho_4 = \rho_5$

Table A3 — Optimal Policies for the Network of Fig. 10
with three different ρ values (see Section 4.1)

ρ_1	ρ_{2-4}	ρ_5	Circuit Thresholds					optimal P_b	uncont. P_b	%† gain
			1	2	3	4	5			
3	0.5	3	2	3	3	3	3	0.595280	0.595422	0.023849
3	0.5	9	2	3	3	3	3	0.757864	0.757978	0.01504
3	1	9	0	3	3	3	3	0.743362	0.747131	0.504463
9	0.1	3	3	3	3	3	3	0.772332	0.772332	0.0

† % gain = $100 (P_b(\text{uncontrolled}) - P_b(\text{optimal})) / P_b(\text{uncontrolled})$

$\rho_{2-4} \Rightarrow \rho_2 = \rho_3 = \rho_4$

Table A4 — Optimal Policies for the Network of Fig. 10 with two different ρ values and increased capacity ($T_i = 8, X_i \leq 6$, see Section 4.1.2)

ρ_1	ρ_{2+}	Γ_Ω	Γ^* (%gain) [†]	$P_b(\Omega)$	$P_b(\Omega^*)$ (%gain) [†]	Ω^*
3	1	6.468452	6.468452 {0.0}	0.075935	0.075935 {0.0}	{6,6,6,6,6, 8,8,8,8,8}
6	1	7.559418	7.559418 {0.0}	0.244058	0.244058 {0.0}	{6,6,6,6,6, 8,8,8,8,8}
6	2	9.596429	9.598803 {0.0247}	0.314541	0.314371 {0.0539}	{5,6,6,6,6, 8,8,8,8,8}
9	0.1	5.431453	5.431453 {0.0}	0.422186	0.422186 {0.0}	{6,6,6,6,6, 8,8,8,8,8}
9	1	7.977183	7.977183 {0.0}	0.386371	0.386371 {0.0}	{6,6,6,6,6, 8,8,8,8,8}
9	2	9.716658	9.742103 {0.2619}	0.428432	0.426935 {0.3494}	{5,6,6,6,6, 8,8,7,8,8}
9	3	11.17181	11.38179 {1.8796}	0.468009	0.458010 {2.1365}	{3,6,6,6,6, 8,8,6,8,8}
9	5	13.51705	14.12021 {4.4622}	0.533895	0.513096 {3.8957}	{2,6,6,6,3, 8,8,3,8,8}
9	7	15.10664	15.91793 {5.3704}	0.591712	0.569786 {3.7057}	{2,6,6,6,2, 8,8,2,8,8}
9	10	16.51665	17.35981 {5.1049}	0.662926	0.645718 {2.5957}	{2,6,6,6,2, 8,8,2,8,8}

[†] Gains are relative to the uncontrolled system.

$\rho_{2+} \Rightarrow \rho_2 = \rho_3 = \rho_4 = \rho_5$

Table A5— Optimal Policies for the nominal (nom) and the improved (imp) networks
(see Section 4.2)

ρ net	Γ_{Ω}	Γ_X {%gain} [†]	Γ^* {%gain} [†]	$P_b(\Omega)$	$P_b(\Omega_X)$ {%gain} [†]	$P_b(\Omega^*)$ {%gain} [†]	Ω_X	Ω^*
2.5 nom	10.1807	10.1811 {0.0040}	10.1811 {0.0042}	0.18555	0.185512 {0.0175}	0.185511 {0.0183}	{5,6,6,6,6}	{5,6,6,6,6, 8,8,7,8,8}
2.5 imp	10.5556	10.55607 {0.0047}	10.55607	0.15555	0.155514 {0.0257}	0.155514	{5,6,6,6,6}	{5,6,6,6,6, 8,8,8}
3.5 nom	12.0336	12.07499 {0.3438}	12.07667 {0.3578}	0.31237	0.310001 {0.7568}	0.309905 {0.7876}	{2,6,6,6,5}	{3,6,6,6,5, 8,8,5,8,8}
3.5 imp	12.6020	12.63151 {0.2340}	12.63151	0.27989	0.278200 {0.6021}	0.278200	{3,6,6,6,5}	{3,6,6,6,5, 8,8,8}
4.5 nom	13.2986	13.48778 {1.4222}	13.50657 {1.5635}	0.40895	0.400543 {2.0555}	0.399708 {2.2598}	{2,6,6,6,3}	{2,6,6,6,4, 8,8,4,8,8}
4.5 imp	13.9609	14.07038 {0.7844}	14.07038	0.37952	0.374650 {1.2824}	0.374650	{2,6,6,6,3}	{2,6,6,6,3, 8,5,8}
5.5 nom	14.2267	14.57842 {2.4721}	14.62406 {2.7929}	0.48267	0.469876 {2.6497}	0.468216 {2.9935}	{1,6,6,6,2}	{2,6,6,6,3, 8,8,3,8,8}
5.5 imp	14.9257	15.19319 {1.7925}	15.19319	0.45725	0.44752 {2.1277}	0.44752	{2,6,6,6,2}	{2,6,6,6,2, 8,4,8}
6.5 nom	14.9364	15.45569 {3.4768}	15.51528 {3.8757}	0.54042	0.524440 {2.9568}	0.522607 {3.2960}	{1,6,6,6,2}	{2,6,6,6,2, 8,8,2,8,8}
6.5 imp	15.6425	16.06403 {2.6946}	16.06403	0.51869	0.505722 {2.5004}	0.505722	{2,6,6,6,2}	{2,6,6,6,2, 8,4,8}
7.5 nom	15.4950	16.13004 {4.0985}	16.23304 {4.7632}	0.58680	0.569865 {2.8860}	0.567119 {3.3541}	{1,6,6,6,1}	{2,6,6,6,2, 8,8,2,8,8}
7.5 imp	16.1936	16.69936 {3.1230}	16.69936	0.56817	0.554684 {2.3736}	0.554684	{2,6,6,6,2}	{2,6,6,6,2, 8,4,8}
8.5 nom	15.9449	16.68042 {4.6127}	16.77403 {5.1997}	0.62483	0.607519 {2.7697}	0.605317 {3.1222}	{1,6,6,6,1}	{2,6,6,6,2, 8,8,2,8,8}
8.5 imp	16.6290	17.17527 {3.2853}	17.17527	0.60873	0.595876 {2.1117}	0.595876	{2,6,6,6,2}	{2,6,6,6,2, 8,4,8}
10 nom	16.4753	17.28274 {4.9006}	17.36501 {5.4000}	0.67049	0.654345 {2.4084}	0.652700 {2.6537}	{1,6,6,6,1}	{2,6,6,6,2, 8,8,2,8,8}
10 imp	17.1321	17.69306 {3.2745}	17.69306	0.65736	0.646139 {1.7068}	0.646139	{2,6,6,6,2}	{2,6,6,6,2, 8,4,8}
15 nom	17.5489	18.34001 {4.5078}	18.39848 {4.8420}	0.76601	0.755467 {1.3770}	0.754687 {1.4787}	{1,6,6,6,1}	{2,6,6,6,2, 8,8,2,8,8}
15 imp	18.1136	18.59585 {2.6621}	18.59585	0.75849	0.752055 {0.8477}	0.752055	{2,6,6,6,2}	{2,6,6,6,2, 8,4,8}

[†] All gains are relative to the uncontrolled system.