A Separation Principle Between Scheduling and Admission Control for Broadband Switching

Jay M. Hyman, *Member, IEEE,* Aurel A. Lazar, *Fellow, IEEE,* and Giovanni Pacifici, *Member, IEEE*

***Abstract*— A framework for joint scheduling and admission control in broad-band switching systems is developed according to a *principle of separation* between these two levels of control. It is shown how an admission control strategy can be tailored to a particular mix of traffic by making use of high-level information from the scheduler. This principle is presented in the context of asynchronous time-sharing (ATS), in which explicit guarantees of cell-level and call-level quality of service (QOS) are given to several traffic classes. The separation principle allows the formulation of an optimal admission control policy, which will maximize the expected system utility while maintaining all QOS guarantees. Several heuristic admission control policies are considered, and are compared against the optimal policy as a benchmark. The *admissible load region* is introduced as a means of quantifying the capacity of a switch under the QOS constraints at the cell and call levels. Numerical calculations for a single MAGNET II switching node carrying two classes of real-time traffic are used to illustrate the effects of different scheduling and admission control policies on both the expected utility and the admissible load region.**

I. Introduction

INTEGRATED telecommunication networks carry traffic of several different classes, including one or more real-time (isochronous) traffic classes, each with its own set of traffic characteristics and performance requirements. Two different approaches have been advanced to deal with this phenomenon. In the circuit-switched approach, sufficient resources are allocated to each call to handle its maximum utilization; this guarantees that the call will get the quality of service (QOS) it requires, but may be wasteful of system resources. In the packet-switched approach, traffic from all sources is packetized, and statistical multiplexing techniques are used to combine all network traffic through a single switching fabric. This allows higher network utilization, but requires more sophisticated controls to ensure that the appropriate QOS is provided.

Asynchronous time sharing (ATS) [1] is a set of resource allocation principles for the design of broad-band packet-switched networks that guarantee quality of service. Intelligent resource allocation techniques are used at all possible contention points to increase network utilization while maintaining QOS guarantees. These principles have broad applicability, and can help to *efficiently* provide QOS in many different network settings. They have already been used in the design of two high-speed integrated networks: MAGNET II [2], a testbed for MAN applications, and TeraNet [3], a Gb/s lightwave network.

ATS-based networks are similar to those based on the asynchronous transfer mode (ATM) [4] in that all traffic offered to the network is in the form of small, fixed-size cells. The primary distinction of ATS is that several classes of traffic with different QOS requirements are considered explicitly at every level of system design, both at the edge and at the core of the network. Therefore, one of the fundamental requirements on ATS systems is that the *core* of the network makes a distinction between traffic classes. The introduction of traffic classes into ATM networks, although not in the ATM standard at this time, may be accomplished in a fully compatible manner. For example, traffic class information could be carried in the Virtual Channel Identifier field of the cell header.

In the current work, we focus in particular on the contention between different traffic classes and the associated controls on two levels. Scheduling mediates the low-level contention for service between cells of different classes, while admission control regulates the acceptance or blocking of incoming traffic on a call-by-call basis. These two levels of control are, of course, related. If too much traffic is allowed to enter the network by an overly lax admission control policy, then no scheduler will be able to provide the requested QOS for all classes. A functioning admission control policy is thus a prerequisite for any guarantee of cell-level QOS. However, our stated goal is not merely to guarantee QOS, but to do so *efficiently,* i.e., without rejecting calls unnecessarily. How should the admission controller determine how strict a policy to adopt? Must it monitor such cell-level phenomena as cell loss rates, burstiness, and the like?

This paper deals with the relationship between the controls on these two levels, and develops a modular framework in which an admission controller can efficiently protect the network from cell-level overloads without direct access to any such low-level measurements. The key principle is that of *separation* between the two levels; the scheduler abstracts high-level statistics from the stream of cell-level events, and presents the admission controller with the concise information it needs. This framework is shown to allow the evaluation and comparison of the overall performance of a system under

Manuscript received April 24, 1992. This work was supported by the National Science Foundation under Grant ECD-88-11111. This paper was presented in part at the Workshop on Distributed Systems: Operations and Management, Santa Barbara, CA, October 1991.

J. M. Hyman was with the Department of Electrical Engineering and the Center for Telecommunications Research, Columbia University, New York, NY 10027. He is now with the Fixed Income Research Department, Lehman Brothers, New York, NY.

A. A. Lazar and G. Pacifici are with the Department of Electrical Engineering and the Center for Telecommunications Research, Columbia University, New York, NY 10027.

IEEE Log Number 9206810.

different suites of scheduling and admission control algorithms.

The paper is organized as follows. Section II provides a brief review of relevant work from the literature. In Section III, the basic concepts of the ATS framework are presented, along with an overview of our architecture for joint scheduling and admission control. In Section IV, the admission control problem is formulated as a constrained maximization of utility, and is shown to be solved by a linear program. Section V describes several heuristic admission control policies whose performance is evaluated, along with a summary of the scheduling policies used. In Section VI, numerical results are presented comparing the various control policies, and the effects of various factors are evaluated quantitatively and qualitatively. In Section VII, several outstanding issues are discussed.

II. Review of Past Work

A review of real-time scheduling algorithms relevant to our work appears in [5]. We will therefore concentrate here only on reviewing past work on admission control.

There has been much research into admission control policies for circuit-switched networks with calls of multiple bit rates. Different admission control strategies may allow different treatment of call requests which may not be admitted into service. Calls may be blocked or queued, or some combination of the two may be employed. Kraimeche [6] examines all three of these possibilities in detail, and investigates many heuristic policies, with an emphasis on the tradeoff between fairness and efficiency. In this paper, the focus on admission control is on pure loss strategies, in which no queueing is employed and all calls are either admitted or immediately blocked. For this case, Kraimeche and Schwartz [7] suggest a class of restricted access policies in which incoming calls are divided into groups with like bit rates, and the total available bandwidth is partitioned among the groups. This class of policies is shown to provide a middle ground between the policies of complete sharing and complete partitioning.

Gopal and Stern [8] consider a similar pure loss system for two classes of traffic with different bandwidth requirements. They formulate the optimal control problem as an unconstrained maximization of expected throughput, and solve it by the dynamic programming method of policy iteration. Ross and Tsang [9] show that more efficient solutions to this problem can be obtained by value iteration. They also apply a methodology of Tijms [10] to reformulate the dynamic programming problem as a linear program; this allows the optimal control policy to be formulated as a maximization of expected throughput under a call blocking constraint which may be considered to enforce fairness. Oda and Watanabe [11] present an alternative linear programming solution, using a methodology similar to that used by Bovopoulos and Lazar [12], [13] in a somewhat different setting. We will show how these admission control techniques, designed for circuit-switched networks, can be successfully applied in a packet-switched environment with cell-level quality of service constraints.

In related work on packet-switched networks, Ferrari and Verma [14] propose a joint scheduling and admission control algorithm for a system with two classes of traffic. They use a version of Earliest Due Date scheduling, along with a priority mechanism. However, the admission control algorithm is perhaps overly pessimistic; to ensure QOS for a "deterministic" class of traffic, which can tolerate no packet loss, the algorithm reserves enough bandwidth for each admitted call to allow it to transmit continuously at its maximum bandwidth, thus foregoing the advantages of statistical multiplexing.

Ferrandiz and Lazar [15] have addressed the admission control problem for sessions of real-time traffic in a packet-switched environment. Analytical methods were used to find the optimal admission control policy under constraints on end-to-end cell delay, average cell loss rate, and average gap length (number of consecutively lost cells). The real-time cell arrival statistics were modeled by a Markov-modulated Poisson process, and the scheduling was assumed to be first-come first-served (FCFS). The optimal policy was found to take the form of a switching surface in the state space. For the more realistic case considered here, with more complex source models and scheduling algorithms, this analytical approach becomes intractable. As yet, there is no direct analytical solution for the optimal admission control policy based on cell- and call-level QOS requirements.

III. Scheduling and Admission Control in the ATS Framework

At the heart of the distinction between ATS and ATM is a clear definition of traffic classes based on QOS considerations; fundamental to any performance analysis is the set of modeling assumptions on which the analysis is based. This section describes these and other key elements of the ATS approach.

In [5], we considered scheduling algorithms for use in networks carrying three classes of traffic. The scheduling task is wholly defined at the cell level: observations consist of quantities such as cell arrivals and queue sizes, while quality of service is measured by cell delays and cell losses. Simulation experiments with call-oriented traffic were used in [5] to define the *schedulable region,* which is the subset of the space of the number of calls for which the cell-level QOS constraints are met. In the current work, a modular approach is pursued, involving cooperation between the processors responsible for scheduling and admission control. Knowledge of the schedulable region, and no other cell-level information, is used at the admission control level to choose a call admission policy. We then develop the concept of the *admissible load region,* which delineates the loading conditions under which QOS can be guaranteed on both the levels of calls and cells.

The definition of three classes of user traffic by distinct QOS constraints is described in Section III-A. Section III-B introduces the traffic models used in this paper on both the cell and call levels. In Section III-C, the scheduling task is described, and the concept of the *schedulable region* is briefly discussed. In Section III-D, it is shown how this concept can provide a bridge between the cell-level QOS requirements of Section III-A and the admission control problem at the call level. In Section III-E, the *admissible load region* is introduced as a quantitative characterization of system performance under a given scheduler and admission controller.

A. Cell- and Call-Level QOS Constraints

The multiclass system model considered in this paper supports four classes of traffic. Three of the traffic classes, Class I, II, and III, transport user traffic, and are defined by a set of performance constraints on the cell as well on the call level. The fourth class, Class C, transports traffic of the network management system, and is not subject to specific QOS constraints. In what follows, we will first define the cell-level constraints.

Class I traffic is characterized by 0% *contention cell loss* and an end-to-end time delay distribution with a narrow support. The maximum end-to-end time delay between the source and destination stations is denoted by S^{I}. Class II traffic is characterized by ϵ% contention cell loss and an upper bound η on the average number of consecutively lost cells. It is also characterized by an end-to-end time delay distribution with a larger support than Class I. The maximum end-to-end time delay is S^{II}. Here, ϵ and η are arbitrarily small numbers and $S^{\mathrm{I}} \leq S^{\mathrm{II}}$. For Class I and II traffic, there is no retransmission policy for lost cells. Class III traffic is characterized by 0% *end-to-end cell loss* that is achieved with an end-to-end retransmission policy for error correction. If requested, it is also characterized by a *minimum* average user throughput Γ and a *maximum* average user time delay T.

The QOS descriptions for each traffic class presented above deal explicitly with the quality of cell-level service guaranteed to all calls admitted to the network. However, this quality of service may be trivially guaranteed by any scheduling mechanism if a conservative admission control policy is used to limit utilization to sufficiently low levels. Guarantees of cell-level QOS to admitted calls is not sufficient if it comes at the cost of unreasonably high rates of call blocking. There is thus a need to simultaneously guarantee a certain quality of service at the call level as well. In this paper, bounds $\kappa^{\mathrm{I}}, \kappa^{\mathrm{II}}$, and κ^{III} on the probability of call blocking for each class define the QOS on the call level.

B. Assumptions about Traffic Statistics

The joint scheduling and admission control problem required us to make assumptions about both the *cell-level statistics* as well as the *call-level statistics*.

In the design of controls for the ATS architecture, we sought robust scheduling algorithms, which would perform well under a wide range of cell arrival statistics corresponding to diverse real-world traffic sources. To this end, a conscious decision was made to eschew the traditional assumption of Poisson cell arrivals in favor of more complex models (see Section V-A). At the call level, however, Poisson call arrivals and exponentially distributed holding times are assumed.

Therefore, each type of service (voice, video, facsimile, etc.) supported by the network will have its own call arrival rate and average holding time, and each call of a given service type will be characterized by the same single-source model or cell interarrival time distribution, which may be arbitrary. In practice, many different service types are likely to be mapped onto a traffic class; in the sequel, however, each traffic class will be assumed for simplicity to consist of calls of a single type of service, with the system state given by a vector $\boldsymbol{x}$ of the integer number of calls of each class.

C. The Scheduler and the Schedulable Region

The task of the scheduler is to resolve contention between cells of different classes at a switching point, such as to satisfy the cell-level QOS requirements for all classes. The scheduler may make use of information from cell-level sensors regarding cell arrivals and departures as well as buffer occupancies (queue lengths). The high speeds at which future integrated networks will operate and the resultant high volume of cell-level information impose real-time constraints on scheduling decisions which mandate the use of simple algorithms with relatively simple information structures. Several different scheduling algorithms with different information structures have been compared in [5], and will be summarized in Section V-B.

If too many calls are admitted to the system, no scheduling algorithm will be able to satisfy the cell-level QOS requirements for all classes. For each scheduling algorithm, experiments involving different numbers of calls of each traffic class are used to determine the boundaries of the state space delineating the set of calls that the system can accept.

This *schedulable region,* denoted by $\mathcal{S}$, has been defined as follows [5]. Let

$$\mathcal{S} = \{\boldsymbol{x} \in \mathcal{N}^n \mid \text{scheduler guarantees the cell-level QOS for all classes}\} \tag{3.1}$$

where $\mathcal{N}$ is the set of natural numbers, and the state $\boldsymbol{x}$, which was informally introduced in Section III-B, will be precisely defined in Section IV-A. The maximum number N^i of Class i calls allowed into the system is defined from the limits of the schedulable region by

$$N^i = \max_{\boldsymbol{x} \in \mathcal{S}} x^i. \tag{3.2}$$

The region $\mathcal{S}$ represents the limits on the admission control policy imposed by QOS considerations at the scheduling level. In general, this region will depend on the statistical characteristics and cell-level QOS constraints of each class of traffic, as well as on the details of the scheduling policy in use. Of necessity, the admission control must take these considerations into account; the approach presented here is to leave as much of these details as possible to be dealt with at the scheduling level, and to pass the minimal necessary amount of information to the admission controller via the delineation of the schedulable region $\mathcal{S}$. The mechanism by which this is accomplished is described in the following section.

D. Joint Scheduling and Admission Control

To achieve efficient use of network resources while guaranteeing QOS, a traffic control architecture known as WIENER

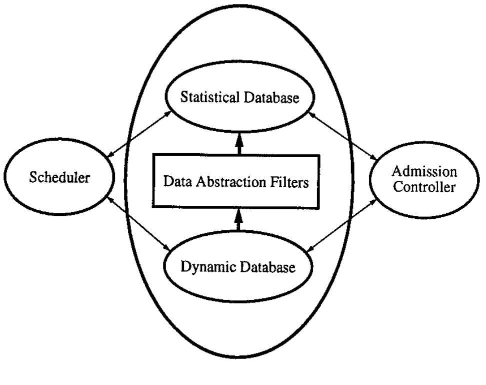


Fig. 1. The architecture for joint scheduling and admission control.

has been developed [16], [17]. Within this architecture, various network management agents operate asynchronously and independently of one another, but can communicate with each other via a common knowledge database to achieve the common goal of optimizing system performance. Although the WIENER framework encompasses routing, flow control, and buffer management as well, we will focus exclusively on scheduling and admission control. The tasks of these two modules, and their interrelationship, are described in this section, and depicted in Fig. 1. The scheduler, as described above, controls the high-speed flow of cells through the switch. The dynamic database temporarily stores the relevant cell-level information for use by the scheduler in making its decisions, as well as the resultant cell-level performance. Meanwhile, data abstraction filters digest this information flow, and continually update a statistical database, which stores various time-averaged and estimated quantities for use by other network entities. Specifically, the schedulable region is stored here for use by the admission controller.

The task of the admission controller is to accept or reject arriving calls so as to maximize a utility function based on the weighted average throughput. It is constrained by the need for the network to guarantee the required QOS at the cell level to all calls admitted into service, and by limits on the call blocking probabilities as well. The information available to the admission controller includes the boundaries of the schedulable region $\mathcal{S}$ specified by the scheduler, the call arrival and departure rates associated with each type of service, and the weights used in the utility function. The admission controller is thus shielded from the flood of detailed cell-level information, and is able to guarantee QOS at the cell level by keeping the number of calls within the schedulable region. The detailed shape of the schedulable region may have a strong impact on the choice of the admission control policy.

We have, therefore, devised a modular framework in which an admission controller can efficiently protect the network from cell-level overloads without direct access to low-level measurements. We call this the *separation principle* between real-time scheduling and admission control. To recap, the scheduler abstracts high-level statistics from the stream of cell-level events, and presents the admission controller with the schedulable region. The latter represents the concise information the admission controller needs for an effective execution of the admission control task. As a formal expression of the goals and limitations of the admission controller, the optimal admission control problem is presented and solved in Section IV. The numerical solutions to this problem can then be used to benchmark the various heuristic control policies presented in Section V-C.

It is worth noting that a similar scheme [18] has been suggested in relation to call routing, which is not addressed here. The scheduler at each contention point in the network registers its schedulable region in a distributed database. The routing control module can then base call routing decisions on the contents of this database, once again insulated from the cell-level performance observations which might have been necessary to monitor the quality of service.

E. The Admissible Load Region

For network service providers concerned with proper dimensioning to meet projected demands, the important question to be asked of a given system is: Under what loading conditions can all QOS constraints be satisfied? The answer to this question may be represented for our multiclass system as a region in the space of call arrival rates of each class. As this region, which we will call the *admissible load region,* is highly dependent on the schedulable region for the switch, as well as the call holding times for the various classes, it provides a very useful characterization of the joint performance of the scheduling and admission control algorithms. It can thus play a role not only in evaluating the merits of one admission control algorithm relative to another under a given scheduler, but also in comparing scheduling algorithms in terms of their impact on final system performance under load.

For a given schedulable region $\mathcal{S}$, call holding times $(\mu^1, \cdots, \mu^n)$, and bounds $(\kappa^1, \cdots, \kappa^n)$ on the call blocking rates, the admissible load region is defined by

$$\begin{aligned}\mathcal{A} = \{(\lambda^1, \cdots, \lambda^n)|\ &\text{admission controller guarantees}\\ &\text{the cell} - \text{ and call} - \text{level QOS for all classes} \quad (3.3)\end{aligned}$$

where the call arrival rates λ^i and holding times μ^i are formally defined in the following section.

IV. The Multiclass Admission Control Problem

In this section, the separation principle presented above is shown to allow a concise mathematical formulation of the optimal admission control policy. This policy will maximize network revenue while guaranteeing both cell-level and call-level QOS. While the call-level QOS constraints appear explicitly in the formulation, the cell-level QOS is guaranteed solely by way of the schedulable region.

Consider a single network node with n classes of traffic. Each traffic class $i \in \{1, \cdots, n\}$ consists of a stream of statistically identical calls with Poisson arrivals at rate λ^i and i.i.d. exponentially distributed holding times with mean $1/\mu^i$. Each traffic class is distinguished by a unique set of performance constraints and per-call traffic characteristics (bandwidth requirements, burstiness, etc.).

The admission control will consist of deciding whether arriving calls should be admitted to the network or whether they should be blocked. (The possibility of queueing incoming calls for later admission is not considered here.) In what follows, a Markov chain model is defined for the system from the point of view of the admission controller, a criterion is defined for the optimality of this control, and the optimal control is formulated as a linear program.

A. The Markov Chain

Let $\boldsymbol{X}_t$ be a continuous-time vector-valued Markov chain, which for any time t is a random variable taking values $\boldsymbol{x} = (x^1, x^2, \cdots, x^n)^T \in \mathcal{S} \subset \mathcal{N}^n$, where $x^i \in \{0, 1, \cdots, N^i\}$ is the number of calls in the system of Class i, and $\mathcal{S}$ and N^i are as defined in (3.1) and (3.2). $\boldsymbol{x}$ represents the state of the Markov chain. Let the admission control policy $\boldsymbol{u}: \mathcal{S} \mapsto \{0,1\}^n$ be defined by the vector mapping $\boldsymbol{u}(\boldsymbol{x}) = (u^1(\boldsymbol{x}), u^2(\boldsymbol{x}), \cdots, u^n(\boldsymbol{x}))^T$, where $u^i(\boldsymbol{x}) = 1$ (0) signifies that the system will accept (reject) an incoming call of Class i when it is in state $\boldsymbol{x}$. Let $\mathcal{U}$ be the set of all possible control policies $\boldsymbol{u}$. Calls of Class i are admitted at a rate λ^i whenever the admission controller allows; each of the x^i Class i calls in service departs at rate μ^i.

The transition rates between two states of the Markov chain $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$ are thus given by

$$q(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} u^i(\boldsymbol{x})\lambda^i & \text{if } y = \boldsymbol{x} + \boldsymbol{e}^i \\ x^i \mu^i & \text{if } y = \boldsymbol{x} - \boldsymbol{e}^i \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where $\boldsymbol{e}^i$ is the elementary vector with a 1 at position i and zeros elsewhere.

Let $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ denote the equilibrium probabilities of the Markov chain, which solve the global balance equations

$$\pi_{\boldsymbol{u}}(\boldsymbol{x}) \sum_{\boldsymbol{y} \in \mathcal{S}} q(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{y} \in \mathcal{S}} \pi_{\boldsymbol{u}}(\boldsymbol{y}) q(\boldsymbol{y}, \boldsymbol{x}), \qquad (\boldsymbol{x} \in \mathcal{S}) \tag{4.2}$$

subject to the normalization equation

$$\sum_{\boldsymbol{x} \in \mathcal{S}} \pi_{\boldsymbol{u}}(\boldsymbol{x}) = 1. \tag{4.3}$$

B. Maximization of Utility

Let the row vector $C = (c^1, \cdots, c^n)$, where the c^i are fixed constants, represent the utility generated by each call of Class i. Many different rules can be used to assign utilities to the different traffic classes. Several examples of possible utility weighting assignments are the following.

- c^i is proportional to the per-call maximum bandwidth required for Class i. This utility assignment corresponds to the maximization of throughput in the circuit-switched case, where each call reserves enough bandwidth for its maximum transmission rate.
- c^i is proportional to the per-call average bandwidth required for Class i. This utility assignment, reflecting the average throughput, is more appropriate in a packet-switched environment, but does not take into account the burstiness of Class i calls.
- c^i is inversely proportional to N^i, the maximum number of Class i calls that may be accepted. This utility assignment most accurately reflects the cost to the network of carrying a call of each class in the ATS framework.

Note that actual utility assignments, or tariffs, will be market-driven, and are thus likely to be more complex, to reflect the effects of many other criteria as well.

The utility generated by the network in state $\boldsymbol{x}$ is $C\boldsymbol{x}$, and the total expected utility $J(\boldsymbol{u})$ under control $\boldsymbol{u}$ is given by

$$J(\boldsymbol{u}) = \sum_{\boldsymbol{x} \in \mathcal{S}} \pi_{\boldsymbol{u}}(\boldsymbol{x}) C\boldsymbol{x}. \tag{4.4}$$

The optimal control policy, if it exists, achieves the maximum utility, i.e.,

$$\max_{\boldsymbol{u} \in \mathcal{U}} J(\boldsymbol{u}). \tag{4.5}$$

As mentioned in Section III-A, there may be a need to guarantee a certain quality of service at the call level, such as a limit on the probability of call blocking. The call blocking probability $p^i(\boldsymbol{u})$ for Class i traffic under a control $\boldsymbol{u}$ is given by

$$\begin{aligned} p^i(\boldsymbol{u}) &= \sum_{\boldsymbol{x} \in \mathcal{S}} (1 - u^i(\boldsymbol{x})) \pi_{\boldsymbol{u}}(\boldsymbol{x}) \\ &= 1 - \sum_{\boldsymbol{x} \in \mathcal{S}} u^i(\boldsymbol{x}) \pi_{\boldsymbol{u}}(\boldsymbol{x}). \end{aligned} \tag{4.6}$$

We therefore add an additional constraint to the formulation of our optimal policy to bound the call blocking probabilities for Class i below a limiting value κ^i:

$$p^i(\boldsymbol{u}) \le \kappa^i. \tag{4.7}$$

The limits κ^i can be different, providing different call-level QOS to the different traffic classes, or they can be the same, thus imposing a *fairness* constraint on the scheduling policy [7].

C. The Optimal Control Policy

The optimal control problem presented in Sections IV-A and IV-B is a nonlinear optimization problem since the equilibrium probabilities $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ obtained as a solution to the balance equations (4.2) contain products of the controls u^i. The methods of dynamic programming (either policy iteration or value iteration) could be applied to find optimal controls for a discounted utility function. However, this approach runs into problems when the utility function is cast as a long-term average, as here formulated. In addition, it does not allow the introduction of the call blocking constraint (4.7). We therefore follow the methodology developed by Bovopoulos and Lazar [12], [13] (also used by Oda and Watanabe [11]) to reformulate the optimization problem as a linear program.

To linearize the problem, we will rewrite the equations of Section IV-A with a new set of variables representing network flows, in addition to the equilibrium probabilities. That is, let

$$\nu^i_{\boldsymbol{u}}(\boldsymbol{x}) = \pi_{\boldsymbol{u}}(\boldsymbol{x}) u^i(\boldsymbol{x}), \qquad (\boldsymbol{x} + \boldsymbol{e}^i \in \mathcal{S}). \tag{4.8}$$

Thus, $\nu_{\boldsymbol{u}}^i(\boldsymbol{x})\lambda^i$ represents the equilibrium flow out of state $\boldsymbol{x}$ due to arrivals of Class i calls. The global balance equations (4.2) for all $\boldsymbol{x} \in \mathcal{S}$ may then be rewritten as

$$\begin{aligned}\sum_{i=1}^{n}(1_{\{\boldsymbol{x}+\boldsymbol{e}^i\in\mathcal{S}\}}\nu_{\boldsymbol{u}}^i(\boldsymbol{x})\lambda^i + \pi_{\boldsymbol{u}}(\boldsymbol{x})x^i\mu^i)\\ = \sum_{i=1}^{n} 1_{\{x^i>0\}}\nu_{\boldsymbol{u}}^i \cdot (\boldsymbol{x}-\boldsymbol{e}^i)\lambda^i\\ + \sum_{i=1}^{n} 1_{\{\boldsymbol{x}+\boldsymbol{e}^i\in\mathcal{S}\}}\pi_{\boldsymbol{u}} \cdot (\boldsymbol{x}+\boldsymbol{e}^i)(x^i+1)\mu^i. \qquad (4.9)\end{aligned}$$

The optimal policy may now be defined by the following linear program in terms of the equilibrium probabilities $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ and the equilibrium arrival flows $\nu_{\boldsymbol{u}}^i(\boldsymbol{x})$:

$$\max \sum_{\boldsymbol{x}\in\mathcal{S}} \pi_{\boldsymbol{u}}(\boldsymbol{x})C\boldsymbol{x}$$

subject to

$$\begin{aligned}&0 \le \pi_{\boldsymbol{u}}(\boldsymbol{x}), \qquad (\boldsymbol{x}\in\mathcal{S})\\ &0 \le \nu_{\boldsymbol{u}}^i(\boldsymbol{x}) \le \pi_{\boldsymbol{u}}(\boldsymbol{x}), \quad (\boldsymbol{x}+\boldsymbol{e}^i\in\mathcal{S}, i\in\{1,\cdots,n\})\\ &\sum_{\boldsymbol{x}\in\mathcal{S}} \pi_{\boldsymbol{u}}(\boldsymbol{x}) = 1\\ &\sum_{\boldsymbol{x}+\boldsymbol{e}^i\in\mathcal{S}} \nu_{\boldsymbol{u}}^i(\boldsymbol{x}) \ge 1-\kappa^i, (i\in\{1,\cdots,n\})\\ &\sum_{i=1}^{n}(1_{\{\boldsymbol{x}+\boldsymbol{e}^i\in\mathcal{S}\}}\nu_{\boldsymbol{u}}^i(\boldsymbol{x})\lambda^i + \pi_{\boldsymbol{u}}(\boldsymbol{x})x^i\mu^i)\\ &\quad = \sum_{i=1}^{n} 1_{\{\boldsymbol{x}+\boldsymbol{e}^i\in\mathcal{S}\}}\pi_{\boldsymbol{u}}(\boldsymbol{x}+\boldsymbol{e}^i)\cdot(x^i+1)\mu^i\\ &\qquad + \sum_{i=1}^{n} 1_{\{x^i>0\}}\nu_{\boldsymbol{u}}^i(\boldsymbol{x}-\boldsymbol{e}^i)\lambda^i\\ &(\boldsymbol{x}\in\mathcal{S}). \qquad (4.10)\end{aligned}$$

The second set of constraints ensures that the controls satisfy $u^i(\boldsymbol{x}) \in [0,1]$. In fact, the properties of the linear program [12], [13] ensure that the solution will satisfy $u^i(\boldsymbol{x}) \in \{0,1\}$ for all (except possibly one) of the control variables $u^i(\boldsymbol{x}), i \in \{1,\cdots,n\}, \boldsymbol{x}\in\mathcal{S}$. The fourth set of constraints is derived from the call blocking constraints (4.7), along with (4.6) and (4.8).

For the special case of the optimal admission control policy, the admissible load region is determined by

$$\begin{aligned}\mathcal{A} = \{(\lambda^1,\cdots,\lambda^n)|&\\ &\text{the linear program (4.10) admits}\\ &\text{a feasible solution}\}. \qquad (4.11)\end{aligned}$$

Note that as the utility weighting vector C appears only in the objective function and not in any of the constraints, it has no effect on determining the admissible load region $\mathcal{A}$, although it will certainly influence the achieved utility.

D. Complexity Considerations

How difficult a problem is this to solve? Is it feasible to consider the optimal admission control for actual implementation in broad-band networks? Let $N = card(\mathcal{S})$, the number of states in the schedulable region. Then the linear program formulated above for n traffic classes has $(n+1)N$ variables and $(n+1)N$ constraints (or $(n+1)N+n$ if call blocking is included). Even for the examples reported in Section VI for $n = 2$, the linear program became quite large, and solutions were accomplished only with difficulty. If we further consider that N itself is exponential in n, we find that it is unrealistic to expect to solve this problem in near-real-time in response to evolving conditions in a broad-band switching system.

Nevertheless, the fact that we can concisely express the optimal solution and solve it numerically for some simple cases can be very useful in several ways. Valuable insights can be gleaned from observations of the form of the optimal policy under various loading conditions and schedulable regions, and can inspire the design of good heuristic control policies. Similarly, the performance results achieved by the optimal control can be used as a benchmark against which all other control policies may be evaluated.

V. Experiments with Two Classes of Real- Time Traffic on a MAGNET II Switching Node

In this section, we describe the experimental procedure used to evaluate the performance of the proposed scheme for joint scheduling and admission control. We will assume that the Ring Switch Fabric of the MAGNET II [2] switching node is congestion-free. Contention at the switching nodes takes place, therefore, only at the output links. Simulations and numerical studies were performed for a single MAGNET II switching link, with a capacity of 100 Mb/s, loaded with real-time traffic of two classes and operated under various control policies at both the scheduling and admission control levels. The traffic sources, the scheduling policies, and the admission control policies considered are described in this section. In all cases, discussion is restricted to Class I and Class II, which are the only classes used here, although the control mechanisms are actually defined for the multiclass traffic model described in Section III-A. (Note that in this section, to reflect the specific nature of the traffic classes considered, we revert to the use of Roman numerals to index the two classes.)

A. Traffic Loading

For the purposes of this paper, we consider two of the traffic classes defined in Section III-A to carry information of a very specific type. Class I is assumed to consist of x^{I} constant bit rate (CBR) video calls, and Class II of x^{II} variable bit rate (VBR) calls. In the determination of the schedulable region (Section V-B), the number of calls is considered constant throughout each experiment; in the investigations of various admission control policies (Section V-C), the number of calls varies according to the Markov model described in Section IV.

A single CBR video call is assumed to generate cells at a constant rate of 1 Mb/s, as in the proposed MPEG standard [19]. A single VBR video call is modeled as a periodic random process characterized by a fixed frame size of duration $F = 62.5$ ms, with cells being generated at a constant rate for a randomly varying portion of each frame. Two different experiments were performed, assuming different levels of burstiness for the VBR video sources. In Experiment 1, the active period of each frame is uniformly distributed between 10 and 40 ms, during which time the source emits cells at a constant rate of 10 Mb/s. For Experiment 2, each VBR source generates cells at the higher rate of 100 Mb/s, but has a correspondingly shorter active period, uniformly distributed between 1 and 4 ms. The average data transmission rate for a single VBR video source is thus approximately 4 Mb/s in either case. This model of VBR video is based on our previous observation of the qualitative behavior of real-time video sequences on MAGNET II [20].

B. Scheduling Policies

In static priority scheduling (SPS), a frequently suggested mechanism for scheduling real-time packet traffic, Class II cells will be transmitted only when there are no Class I cells awaiting transmission. This scheduling mechanism offers the best possible service to Class I, but can cause long delays for Class II traffic.

In asynchronous time sharing (ATS), communication resources are time-shared between the classes according to a cycle scheme [1]. The MAGNET II Real-Time Scheduling (MARS) Algorithm [5] is a mechanism for adaptively setting the parameters which govern this cycle scheme, based on observations of cell arrivals and departures. The algorithm allocates to Class I the minimum amount of resources it needs to satisfy its QOS requirements. Thus, Class I cells may be delayed slightly (within their deadlines) in order to improve performance for Class II.

The schedulable regions for these two scheduling algorithms were determined by simulation experiments as in [5] for the two classes of traffic described above. The QOS constraints described in Section III-A were imposed for Classes I and II, using the QOS vector $[S^{\mathrm{I}}, S^{\mathrm{II}}, \epsilon, \eta]$ = [1 ms, 1 ms, 0.001, 5.0]. We let Class I traffic vary from $x^{\mathrm{I}} = 0$ CBR video calls to $x^{\mathrm{I}} = 100$ calls. For each simulation run, we fixed x^{I} and determined the maximum number of VBR video calls x^{II} for which the cell-level QOS constraints could be satisfied. The schedulable regions for these two algorithms are plotted against the regions corresponding to circuit switching (CKT) in Fig. 2, using the VBR source models of Experiments 1 and 2. Note that MARS can accommodate a significantly higher number of CBR sources when the system is highly loaded with VBR traffic. Also note the great advantage of both packet-switched schedulers (SPS and MARS) over circuit switching, especially when using the relatively smooth traffic sources of Experiment 1.

C. Admission Control Policies

In this section, several heuristic admission control policies are presented for comparison. Each of these policies in turn will be evaluated using the two-class system and the MARS scheduling algorithm, and the results compared to those achieved by the optimal policy.

The Complete Sharing Policy: The complete sharing (CS) policy always admits incoming calls, except in cases where doing so would lead to a state outside of the schedulable region. That is,

$$u^i(\boldsymbol{x}) = 1_{\{\boldsymbol{x}+\boldsymbol{e}^i \in \mathcal{S}\}}. \tag{5.1}$$

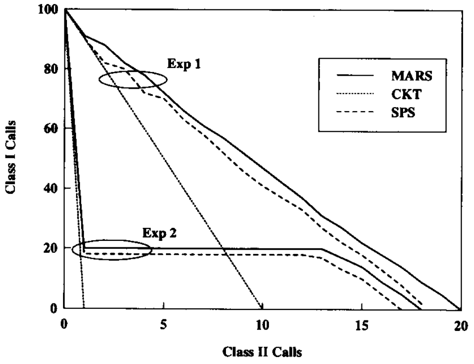


Fig. 2. Class I and II maximum number of calls, QOS = [1 ms, 1 ms, 0.001, 5.0].

This situation may also be referred to as the uncontrolled case, from the point of view of the admission controller. When the CS policy is used, the schedulable region $\mathcal{S}$ specified by the scheduler completely determines whether an incoming call will be admitted or blocked.

The Hold-Back-One Control Policy: While the CS policy given above may seem intuitive at first, there may be situations in which it is not the best policy. Specifically, if calls of one class need significantly less bandwidth than others, it may be desirable to reject a call of this type which could have been accommodated, to reduce the likelihood of future rejection of a larger call.

As an example of a control policy exhibiting this type of behavior, consider the following heuristic policy for the two-class situation in which calls of Class II need considerably more bandwidth than calls of Class I:

$$\begin{aligned} u^{\mathrm{I}}(\boldsymbol{x}) &= 1_{\{\boldsymbol{x}+\boldsymbol{e}^{I} \in \mathcal{S}\}} \wedge \neg(1_{\{\boldsymbol{x}+\boldsymbol{e}^{II} \in \mathcal{S}\}} \wedge 1_{\{\boldsymbol{x}+\boldsymbol{e}^{II}+\boldsymbol{e}^{I} \notin \mathcal{S}\}}) \\ u^{\mathrm{II}}(\boldsymbol{x}) &= 1_{\{\boldsymbol{x}+\boldsymbol{e}^{II} \in \mathcal{S}\}}. \end{aligned} \tag{5.2}$$

This algorithm tries to avoid the situation in which Class II calls and Class I calls could be accepted, but accepting an arriving Class I call would cause subsequent Class II calls to be rejected. A slot is thus held open for the anticipated future arrival of a Class II call (see Fig. 4). A similar heuristic was considered by Kraimeche [6] for the multirate circuit-switched problem.

The Multidimensional Threshold Control Policy: The multidimensional threshold (MDT) policy is designed to achieve the blocking probability constraints for each class by the imposition of thresholding rules. Let z be the thresholding point; then define

$$u^i(\boldsymbol{x}) = 1_{\{x^i < z^i\}} \wedge 1_{\{\boldsymbol{x}+\boldsymbol{e}^i \in \mathcal{S}\}}. \tag{5.3}$$

If $z \in \mathcal{S}$, this policy takes the form of a complete partitioning (CP) policy. (Note that the CP policy is different from the CS policy previously defined.) Each class, in effect, has its own dedicated bandwidth, and no call-level interference between

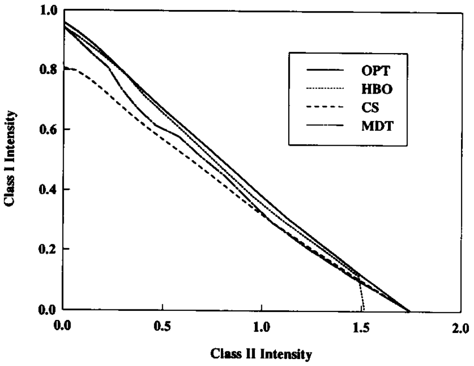


Fig. 3. Admissible load regions for the optimal, HBO, MDT, and complete sharing admission control policies, MARS scheduling, smooth VBR video sources, and call blocking constraints $\kappa^{\mathrm{I}} = \kappa^{\mathrm{II}} = 0.1$.

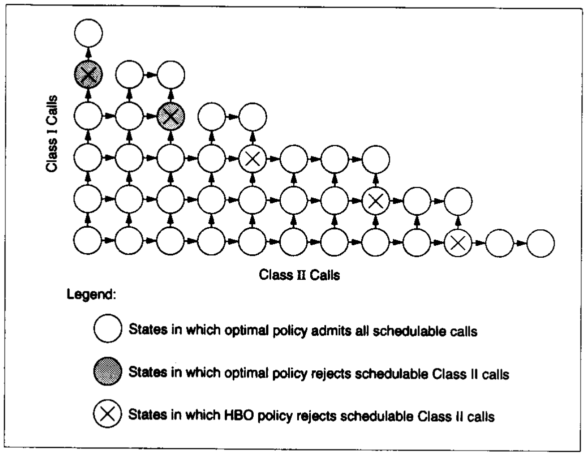

Class I Calls
Class II Calls
Legend:
States in which optimal policy admits all schedulable calls
States in which optimal policy rejects schedulable Class II calls
States in which HBO policy rejects schedulable Class II calls

Fig. 4. Illustration of the optimal and HBO control policies.

classes takes place. In this case, the blocking probabilities p^i for each class are given exactly by

$$p^i(\boldsymbol{u}) = E(\lambda^i/\mu^i, z^i) \tag{5.4}$$

where $E(A, N)$ is the one-dimensional Erlang-B formula:

$$E(A, N) = \frac{A^N/N!}{\sum_{i=0}^{N} A^i/i!}. \tag{5.5}$$

This observation has led us to choose our thresholding point $\boldsymbol{z}$ such as to try to reserve for class i the number of dedicated servers, it would need in a CP setting,

$$z^i = \min_{E(\lambda^i/\mu^i, N) < \kappa^i} N. \tag{5.6}$$

If, for $\boldsymbol{z}$ so derived, we have $\boldsymbol{z} \in \mathcal{S}$, then this policy will be a CP policy, and it will guarantee call-level QOS for all classes. (In this case, $\boldsymbol{z}$ may be increased to the boundary of $\mathcal{S}$ by any algorithm desired and in any direction without violating this guarantee. This amounts to a choice among a set of CP policies which all guarantee QOS, but may provide different payoffs for the specific values of λ and μ in effect.)

When $\boldsymbol{z} \notin \mathcal{S}$, the policy defined by (5.3) is no longer equivalent to the CP policy since the boundary of $\mathcal{S}$ may now cause some calls of each class to be blocked even before hitting the threshold z^i. The blocking rates for each class will thus exceed those given by the one-dimensional approximation of (5.4). Nevertheless, it is often the case that this increase in blocking is small enough such that each class still meets its blocking probability constraint.

The intuitive appeal of this heuristic is that it tends to bring the actual blocking probabilities for all classes fairly close to their specified limits. In cases where the CS policy results in QOS violations for one class while other classes remain lightly loaded, this thresholding policy tends to increase blocking for the lightly loaded classes and to decrease it for the class experiencing excessive loading. Compared to the HBO policy, note that HBO is a fixed policy defined for our two-dimensional case based on an *a priori* notion of a favored traffic class. It is not entirely clear how it should be extended to handle more than two traffic classes. The MDT policy, however, is well defined for any number of classes. In addition, its decisions on which classes to favor and which to block are based not made *a priori* as part of the policy description, but are dynamically derived based on the current loading conditions and blocking constraints for each class.

VI. Numerical Results

The performance of the various joint scheduling and admission control policies was evaluated both qualitatively and quantitatively for the two-class model system described in Section V. Experiments were conducted using the six different scheduling regions pictured in Fig. 2, using MARS, SPS, and circuit switching with two different sets of VBR source model parameters.

Two distinct types of gain can be identified: an increase in the admissible load region, allowing the system to operate at higher offered loads; and an increase in utility at a given offered load. While these two types of gain are not mutually exclusive, we have found the former type of gain to be the dominating effect when call blocking constraints are imposed. When these constraints are relaxed, the latter type of gain becomes prominent.

In the majority of this section, we focus on the admissible load region and its dependence on the admission control policy, the scheduling policy, the traffic statistics, and the QOS constraints. In Section VI-A, the size and shape of the admissible load region is quantified for different admission control policies. In Section VI-B, the optimal admission controls are applied using different schedulable regions to obtain the admissible load regions achieved by MARS, SPS, and circuit switching, and to highlight the effect of different traffic source models. In Section VI-C, the admissible load regions are compared for different values of the call-level QOS parameters κ^i.

In Section VI-D, we turn our attention to the utility function. The call blocking constraints are relaxed to allow a qualitative and quantitative comparison of the optimal admission control policy with the heuristic control policies over a broad range of intensities. Note that while all of the results on the admissible load region are independent of the utility weights assigned to calls of different classes, as observed in Section IV-C, the utility functions obtained by the different control schemes show a very strong dependence on these weights.

Traffic intensity was parametrized using the Erlang model commonly used for circuit switching. Thus, given that a maximum of $\mathcal{N}^i$ calls of Class i can be accommodated by the circuit-switching approach, the Class i traffic intensity (in Erlangs per server) is given by

$$\rho^i = \frac{\lambda^i}{\mathcal{N}^i \mu^i}. \tag{6.1}$$

The total system traffic intensity can then be defined by

$$\rho = \rho^{\mathrm{I}} + \rho^{\mathrm{II}} \tag{6.2}$$

and the percentage of traffic intensity due to Class I is

$$\frac{\rho^{\mathrm{I}}}{\rho^{\mathrm{I}} + \rho^{\mathrm{II}}} = \frac{\rho^{\mathrm{I}}}{\rho}. \tag{6.3}$$

Throughout this section, the admissible load regions were calculated as follows. The values of μ^{I} and μ^{II} were fixed at 1.0, while λ^{I} and λ^{II} were proportionately raised to increase the total traffic intensity ρ while keeping the ratio ρ^{I}/ρ constant. For the optimal control policy, the linear program was solved for each parameter set, and the intensity ρ was increased in this way until the linear program found no feasible solution. For each heuristic control policy, the global balance equations (4.2) were solved for each parameter set, and the Class I and II call blocking probabilities p^{I} and p^{II} were computed by (4.6). The intensity ρ was then increased until a violation of call-level QOS occurred. This procedure was repeated for 15 values of ρ^{I}/ρ to identify each admissible load region.

A. Effect of the Admission Control Policy on the Admissible Load Region

In the first set of calculations, we investigate the extent to which the optimal control policy allows the system to be operated under a higher call intensity than would be permissible under the various heuristic admission control policies. Fig. 3 shows the admissible load regions achieved by the CS, HBO, MDT, and optimal admission control policies using MARS scheduling, the VBR source model of Experiment 1, and call blocking constraints $\kappa^{\mathrm{I}} = \kappa^{\mathrm{II}} = 0.1$.

The CS policy is seen to underperform the optimal control by more than 15% when Class I traffic dominates. The HBO policy does almost as well as the optimal policy in this case, but performs very poorly when Class II traffic dominates. The MDT policy does not lead to such extreme drops in performance no matter which traffic class dominates, but rather performs moderately well across the entire range of values. A qualitative explanation of these results follows.

In a situation where one class of traffic is experiencing call blocking probabilities near its allowed limit while the other class has relatively low call blocking rates, offered loads can be increased by decreasing the blocking rates for the class near the limit. This can be achieved by blocking more calls of the other classes.

With the specific traffic sources used here, the Class I blocking probability was an order of magnitude smaller than that of Class II under the CS policy since the larger calls of Class II were much more often blocked. The HBO policy (which here was applied to always leave room for a Class II call) was found to always increase the blocking probability for Class I, while decreasing the blocking probability for Class II. This generally had the effect of making the call blocking probabilities for the two classes approximately equal, allowing the load to be raised to a higher value before either of the blocking constraints were violated. As discussed in Section V-C, the MDT policy is designed to achieve exactly this effect. However, it produces somewhat erratic results, as reflected in the uneven shape of its admissible load region in Fig. 3. This inconsistency represents the effect of choosing an integer threshold to bound the real-valued blocking probability; this bound may be looser or tighter depending on the precise loading conditions, leading to variations in how successful the policy is at meeting its goals.

Note that the limiting values as Class II intensity goes to zero do not converge, as might be expected, to a single point corresponding to the single-class case. This is due to the fact that even as Class II intensity approaches zero, the blocking probability constraint for Class II is still imposed, thus limiting the Class I intensity that may be allowed.

The optimal policy, as expected, provides the largest admissible load region of the three. One interesting result that emerged from our studies was that the form of the optimal policy was often similar to the HBO policy. That is, the optimal policy tends to reject Class I calls in the area near the boundary for Class II schedulability. Fig. 4 illustrates the various control policies for a typical case, shown for a system with a much smaller state space to allow a pictorial representation. (In this figure, Class I calls are assumed to be the larger calls, and Class II calls are therefore blocked.) The arrows mark upward transitions allowed by the optimal control policy. (All downward transitions are allowed, but are not shown.) At the two shaded states, Class II calls will be rejected. HBO policy would reject Class II calls at these two states, as well as three others. Note that these two policies exhibit "holes"; that is, there are states $(x^{\mathrm{I}}, x^{\mathrm{II}})$ (the shaded states and those marked "×" in the figure) such that $u^{\mathrm{II}}(x^{\mathrm{I}}, x^{\mathrm{II}}) = 0$, but $u^{\mathrm{II}}(x^{\mathrm{I}}, x^{\mathrm{II}} + 1) = 1$.

B. Effect of the Schedulable Region on the Admissible Load Region

Since the admissible load region reflects the satisfaction of both the call-level and the cell-level QOS constraints, a change in the schedulable region will of necessity affect the admissible

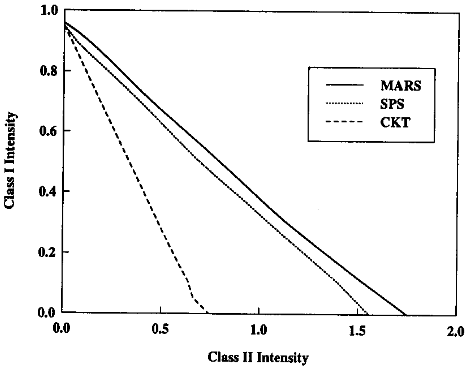


Fig. 5. Admissible load regions for circuit switching, MARS scheduling, and SPS scheduling using the optimal admission control policy, smooth VBR video sources, and call blocking constraints $\kappa^{I} = \kappa^{II} = 0.1$.

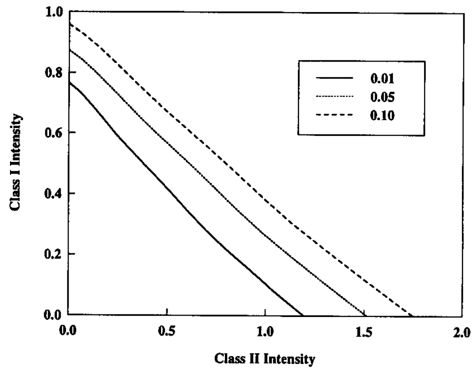


Fig. 7. Admissible load regions for three values of $\kappa^{I} = \kappa^{II}$ using the optimal admission control policy, MARS scheduling, and smooth VBR video sources.

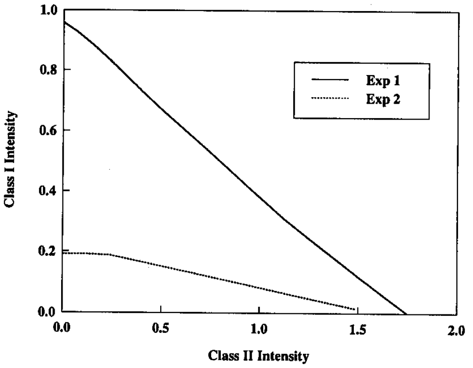


Fig. 6. Admissible load regions using the VBR video sources of Experiments 1 and 2 with MARS scheduling, optimal admission control, and call blocking constraints $\kappa^{I} = \kappa^{II} = 0.1$.

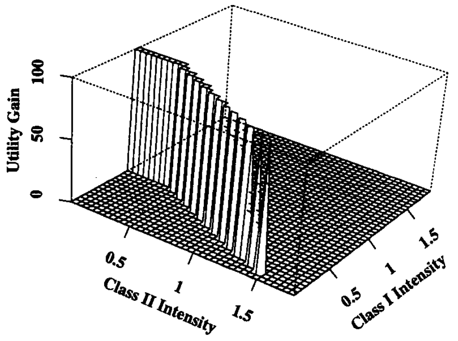


Fig. 8. Utility gain function for the optimal policy over CS, MARS scheduling, smooth VBR video sources, call blocking constraints $\kappa^{I} = \kappa^{II} = 0.1$, $C = (1,4)$.

load region. The dependence of the schedulable region on the scheduling algorithm, the cell-level QOS constraints, and the traffic statistics was explored in detail in [5].

In this section, we show the effect of these dependencies on the admissible load region. The optimal admission control policy is always used for admission control, but with different schedulable regions corresponding to different policies and/or traffic statistics at the scheduling level. It is thus possible to quantify, e.g., the effect on performance at the call level due to the use of the MARS algorithm at the scheduling level.

Fig. 5 shows the admissible load regions achieved by the MARS and SPS scheduling algorithms, compared with that for circuit switching, using the optimal admission control policy, the VBR source model of Experiment 1, and call blocking constraints $\kappa^{I} = \kappa^{II} = 0.1$. Fig. 6 shows the dependence of the admissible load region on the parameters of the VBR video source model.

These plots show the usefulness of the admissible load region as a tool for evaluating cell-level phenomena, which at first may appear inappropriate. The comparison of different scheduling algorithms by the size of their schedulable regions, as in Fig. 2, is not sufficient to quantify the gain of one scheduling algorithm over another. The effective advantage afforded by a larger schedulable region will depend on how likely the system is to operate in that region. This, in turn, will depend on the call arrival and departure statistics, as well as the admission control policy in effect.

C. Effects of Call-Level QOS on the Admissible Load Region

The third major factor in determining the admissible load region is the call-level QOS specification, i.e., the call blocking constraints. Fig. 7 shows the admissible load regions obtained using three different values for the call blocking parameters κ.

D. Utility Gain

In addition to exploring the boundaries of the admissible load region, it is also instructive to evaluate the gain in utility achieved by imposing various controls. In the cases we have examined, utility increases are mainly due to extending the admissible load region to yield higher utility. The utility gain offered for the experiment of Fig. 3 is pictured in Fig. 8. The utility weighting vector $C = (c^{I}, c^{II})$ was assigned values based on the per-call average bandwidth for each class.

Fig. 9 shows the utility function for the CS policy with no call blocking constraints, using MARS scheduling, the smooth VBR video sources of Experiment 1, and a utility weighting vector $C = (1, 12)$. This weighting reflects an assumption

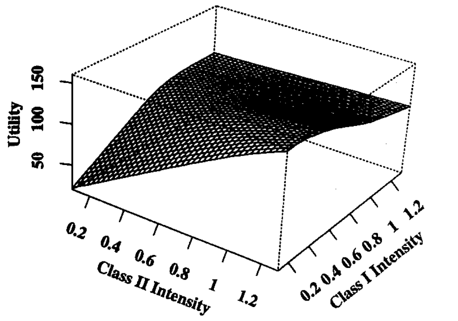


Fig. 9. Utility function for the CS policy, MARS scheduling, smooth VBR video sources, no call blocking constraints, $C = (1, 12)$.

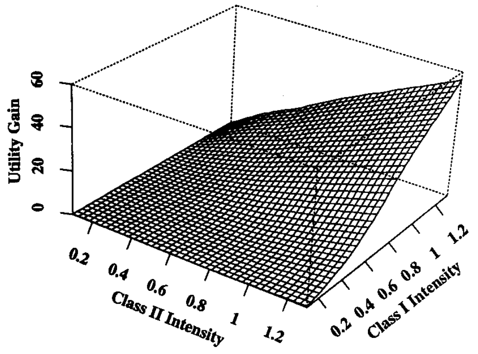


Fig. 10. Utility gain function for the optimal policy over CS, MARS scheduling, smooth VBR video sources, no call blocking constraints, $C = (1, 12)$.

that the Class II calls were three times as valuable (per unit bandwidth) as the Class I calls. Fig. 10 shows the utility gain of the optimal policy over the CS policy in this case. When the utility was considered proportional to the average bandwidth, similar effects were observed; however, the magnitude of the gain was smaller.

In general, it was found that under low to moderate loading conditions, the CS policy was nearly optimal. At higher offered loads, however, the CS policy suffers from an instability effect previously observed by Kraimeche and Schwartz [7]. As the offered loads are increased, there is a sudden increase in the blocking for Class II calls, leading to a drop in utility. The HBO policy (not shown) ameliorates this problem somewhat, but can suffer from the same problem as loads continue to increase. The optimal control, by avoiding this instability, is able to achieve significant gains in utility. At high loads, the optimal control can be seen to achieve a utility gain greater than 60%. This gain is particularly evident when Class II calls were considered to be more valuable, as reflected in the utility weighting vector C. By rejecting more of the less profitable calls (even when they could have been admitted) in anticipation of arrivals of the more expensive calls, the latter need to be blocked less often. The optimal control, therefore, is very strongly dependent on the utility vector C which defines the relative worth of calls of different classes.

The MDT policy was not considered in this section, as its formulation was based on the call blocking probability constraint, which was relaxed here. That policy, as formulated, is therefore not applicable. However, the corresponding policy in the case of no blocking constraint would be to choose some z on the boundary of the schedulable region, which would then define a CP policy as discussed. The choice of which of the possible CP policies to implement, like the choice of the optimal control, will be influenced by the utility vector C. Any reasonable choice of a CP policy, however, will avoid the instability suffered by the CS policy under heavy loading conditions [7].

VII. Conclusion

The problem of joint scheduling and admission control for broad-band switching is difficult to analyze because the cell-level arrival (and departure) statistics are not Markovian. We have been able to gain an insight into this class of problems by limiting the analysis to a basic multiplexing unit and by invoking a separation principle between the two levels of controls. Under this principle, the cell-level information available at the scheduler is presented as a stability (schedulable) region to the admission controller. Within the stability region, cell-level QOS requirements are satisfied. The schedulable region then appears as a constraint on the admission control problem that is assumed to have Markovian statistics. The separation principle defined and used in this paper is closely related to separation theorems derived in a more formal setting in [21] and [22].

The examples in this paper have been limited to the case of a single node, with only two types of calls. By restricting our study to a smaller problem which may be fully solved, we have been able *to gain an understanding* into the behavior of the optimal admission control policy, and when various heuristics can approximate its performance. How can this be extended to the problem of admission control in a large integrated network supporting many types of services? The separation principle enables us to extend the conceptual analysis to multiple switching nodes that do not necessarily use the same scheduling mechanism. In effect, any scheduler can be used provided that the schedulable region is made available to the admission controller.

Adding services and network nodes corresponds to exponential growth in the size of the linear program we have formulated. Clearly, due to the computational constraints described above, we cannot hope to solve for the optimal policy for networks of realistic size, and heuristic policies must be chosen. However, even the much simpler problem of calculating the blocking probabilities for the CS and MDT policies, which admit product-form solutions for the equilibrium probabilities, quickly becomes impractical to solve. It thus will be difficult even to evaluate the performance of various heuristic policies in large networks.

Acknowledgment

The authors would like to thank the reviewers for their constructive comments.

References

[1] A. A. Lazar, A. T. Temple, and R. Gidron, "An architecture for integrated networks that guarantees quality of service," *Int. J. Digital Analog Commun. Syst.*, vol. 3, pp. 229–238, Apr.–June 1990.

[2] A. A. Lazar, A. T. Temple, and R. Gidron, "MAGNET II: A metropolitan area network based on asynchronous time sharing," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1582–1594, Oct. 1990.

[3] R. Gidron and A. T. Temple, "TeraNet: A multihop multichannel lightwave network," in *Proc. IEEE Int. Conf. Commun.*, Denver, CO, June 1991, pp. 602–608.

[4] M. de Prycker, *Asynchronous Transfer Mode: Solutions for Broadband ISDN.* New York: Ellis Horwood, 1991.

[5] J. M. Hyman, A. A. Lazar, and G. Pacifici, "Real-time scheduling with quality of service constraints," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 1052–1063, Sept. 1991.

[6] B. Kraimeche, "Traffic access controls for integrated networks," Ph.D. dissertation, Columbia Univ., New York, NY, 1984.

[7] B. Kraimeche and M. Schwartz, "Circuit access control strategies in integrated digital networks," in *Proc. IEEE INFOCOM,* San Francisco, CA, Apr. 1984, pp. 230–235.

[8] I. S. Gopal and T. E. Stern, "Optimal call blocking policies in an integrated services environment," in *Proc. 17th Conf. Inform. Sci. Syst.*, The Johns Hopkins Univ., Baltimore, MD, 1983.

[9] K. W. Ross and D. H. K. Tsang, "Optimal circuit access policies in an ISDN environment: A Markov decision approach," *IEEE Trans. Commun.*, vol. 37, pp. 934–939, Sept. 1989.

[10] H. C. Tijms, *Stochastic Modelling and Analysis: A Computational Approach.* New York: Wiley, 1986.

[11] T. Oda and Y. Watanabe, "Optimal trunk reservation for a group with multislot traffic streams," *IEEE Trans. Commun.*, vol. 38, pp. 1078–1084, July 1990.

[12] A. D. Bovopoulos and A. A. Lazar, "Optimal routing and flow control of a network of parallel processors with individual buffers," in *Proc. 23rd Annu. Allerton Conf. Commun., Contr., Comput.*, Oct. 1985, pp. 564–573.

[13] A. D. Bovopoulos and A. A. Lazar, "Optimal resource allocation for Markovian queueing networks: The complete information case," *Stochastic Models,* vol. 7, no. 1, 1991.

[14] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 368–379, Apr. 1990.

[15] J. M. Ferrandiz and A. A. Lazar, "Admission control of real-time sessions of an integrated node," in *Proc. IEEE INFOCOM,* Bal Harbour, FL, Apr. 1991, pp. 553–559.

[16] A. A. Lazar, "A real-time control, management and information transport architecture for broadband networks," in *1992 Int. Zurich Seminar Digital Commun.*, Zurich, Switzerland, Mar. 1992, pp. 281–296.

[17] S. Mazumdar and A. A. Lazar, "Monitoring integrated networks for performance management," in *Proc. IEEE Int. Conf. Commun.*, Atlanta, GA, Apr. 1990, pp. 289–294.

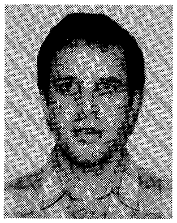
[18] C. Courcoubetis, G. Fouskas, A. A. Lazar, S. Leventis, and S. Sartzetakis, "WIENER and NEMESYS: A comparison of two quality-of-service network management experiments," in *Proc. 4th RACE Telecommun. Management Network Conf.*, Dublin, Ireland, Nov. 1990.

[19] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, pp. 47–58, Apr. 1991.

[20] A. A. Lazar, G. Pacifici, and J. S. White, "Real-time traffic measurements on MAGNET II," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 467–483, Apr. 1990.

[21] F. Vakil and A. A. Lazar, "Flow control protocols for integrated networks with partially observed voice traffic," *IEEE Trans. Automat. Contr.*, vol. AC-32, pp. 2–14, Jan. 1987.

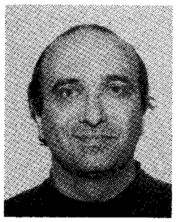
[22] M. T. Hsiao and A. A. Lazar, "Optimal decentralized flow control of Markovian queueing networks with multiple controllers," *Performance Eval.*, vol. 13, pp. 181–204, Nov. 1991.



Jay M. Hyman (S'82–M'92) was born in New York, NY, on June 26, 1961. He received the B.S., M.S., and M. Phil. degrees in 1983, 1984, and 1987, respectively from the Department of Electrical Engineering, Columbia University, New York, NY.

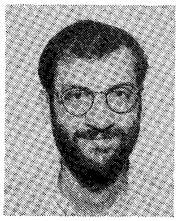
As a Graduate Research Assistant for several years in Columbia's Center for Telecommunications Research, he participated in research projects on speech processing and neural networks. He is nearing completion of his doctoral studies in the areas of scheduling and admission control for integrated telecommunication networks. He is currently employed as a Vice President in the Fixed Income Research Department of Lehman Brothers, New York, NY, where he works on pricing models and applications in the realm of financial securities.

Mr. Hyman is a member of Tau Beta Pi and Eta Kappa Nu.



Aurel A. Lazar (S'77–M'80–SM'90–F'93) has been a Professor of Electrical Engineering at Columbia University, New York, NY, since 1988. His research interests span both theoretical as well as experimental studies of telecommunication networks and intelligent systems. The theoretical research he conducted during the 1980's pertains to the modeling, analysis, and control of telecommunication networks and intelligent systems. He formulated optimal flow and admission control problems and, by building upon the theory of point processes, derived control laws for Markovian queueing network models in a game-theoretic setting. He was the chief architect of two experimental networks, generically called MAGNET. This work introduced traffic classes with explicit quality of service constraints to broad-band switching, and led to the concepts of schedulable and admissible load regions for real-time control of broad-band networks. He is currently working on the foundations of the control and management architecture of future giant gigabit networks. He is the Director of the Telecommunication Networks Laboratory of the Center for Telecommunications Research at Columbia.

Dr. Lazar is Area Editor for Network Management and Control for the IEEE Transactions on Communications, Editor of the *ACM/Springer Verlag Multimedia Systems Journal,* a member of the editorial board of *Telecommunications Systems,* and Editor of the Springer-Verlag monograph series, *Telecommunication Networks and Computer Systems.*



Giovanni Pacifici (S'80–M'85) was born in Rome, Italy, on September 27, 1957. He received the Laurea and Research Doctorate degrees from Department of Information and Communication Technology, University of Rome La Sapienza in 1984 and 1989, respectively.

As a student, his main activities were focused on the performance evaluation of local and metropolitan area networks, with an emphasis on the integration of voice and data. In the course of his studies, he was a Visiting Scholar at the Center for Telecommunications Research, Columbia University, from 1987 to 1988. In 1989 he joined the staff of the Center for Telecommunications Research as a Research Scientist. His current interests include network management architectures and integration of management and control of broad-band integrated networks.