Title	Bandwidth Allocation By Pricing In ATM Networks
Authors(s)	Murphy, Liam, B.E.; Murphy, John
Publication date	1994-03
Publication information	IFIP Transactions C: Communications Systems, C-24 : 333-351
Publisher	Elsevier
Link to online version	https://www.elsevier.com/books/broadband-communications-ii/tohme/978-0-444-81834-8
Item record/more information	http://hdl.handle.net/10197/7625
Publisher's statement	þÿThis is the author s version of a work that was accepted fo C: Communications Systems. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in IFIP Transactions C: Communications Systems (VOL -24, ISSUE 1994, (1994)) DOI:#.

Bandwidth Allocation By Pricing In ATM Networks

John Murphy[a] and Liam Murphy[b]

[a]Electronics Engineering Department
Dublin City University
Ireland

[b]EECS Department
University of California, Berkeley
USA

Admission control and bandwidth allocation are important issues in telecommunications networks, especially when there are random fluctuating demands for service and variations in the service rates. In the emerging broadband communications environment these services are likely to be offered via an ATM network. In order to make ATM future safe, methods for controlling the network should not be based on the characteristics of present services. We propose one bandwidth allocation method which has this property. Our proposed approach is based on pricing bandwidth to reflect network utilization, with users competing for resources according to their individual bandwidth valuations. The prices may be components of an actual tariff or they may be used as control signals, as in a private network. Simulation results show the improvement possible with our scheme versus a leaky bucket method in terms of cell loss probability, and confirm that a small queue with pricing can be efficient to multiplex heterogeneous sources.
Keyword Codes: C.2.1; C.2.4
Keywords:Network Architecture and Design; Distributed Systems

1. INTRODUCTION

Asynchronous Transfer Mode (ATM) is a form of fast packet switching which uses short fixed–length packets [1]. ATM has been adopted by CCITT as the transfer mode for the Broadband Integrated Services Digital Network (BISDN), a service–independent network capable of supporting all the communication services that users now require or may require in the future. ATM is also emerging as a local area networking technology, as it provides flexible bandwidth–on–demand and internetworking capabilities for conventional data communications. Although ATM was originally developed with high bandwidth optical fiber as the intended transmission medium, there is increasing interest in using ATM for lower bandwidth non–fiber environments such as satellite or wireless networks [2], [3]. If ATM can in fact transport any service – irrespective of its bit rate, bursty nature or quality requirements – it offers the possibility of a unifying communication paradigm, allowing not simply integrated applications but also integrated operating environments and administrative domains [4].

ATM is connection–oriented and currently provides no link–by–link error protection or flow control within a connection, called a virtual circuit. The fixed size and reduced header functionality of ATM cells is intended to facilitate the very high–speed switching necessary in networks with link speeds of 150–600 Mbps. The short cell length (48 byte information field + 5 byte header) is intended to keep the delays for real–time services such as voice within acceptable limits. However ATM is a relatively new concept and its implementation is still a very dynamic area, with a wide range of proposed solutions and little practical experience with actual users. Whether ATM can live up to its promise as a universal transfer mode is unknown and many issues have yet to be decided [5].

CALL ADMISSION CONTROL

One of the fundamental issues which has not yet been satisfactorily resolved is the mechanism by which user traffic will be accepted into the network, usually referred to as Call Admission Control (CAC). When a new connection is requested by a user, the network must decide whether or not to accept the call; and if so, how to route it through the network and what resources to reserve for its virtual circuit. Packet–switched networks use higher–layer protocols to guarantee acceptable packet delivery but these are not expected to scale well to broadband speeds. In circuit–switched networks (such as most telephone networks) the CAC mechanism results in call blocking when the bandwidth of a requested connection exceeds the available bandwidth [6]. But in an integrated–services network the traffic source may be *bursty*, so the required bandwidth of its virtual circuit varies with time during the call. The nature of this time–varying behavior and the mean bandwidth requirement vary widely among different sources. Therefore it is difficult to characterize the bandwidth of a requested connection. This difficulty has led to proposals to reserve the peak bandwidth of the call (deterministic multiplexing), as required for constant bit rate (CBR) sources. However the gain in efficiency possible by taking advantage of the statistical nature of variable bit rate (VBR) sources has led to many schemes for statistical multiplexing. Such schemes assign less than the peak bandwidth required, and therefore may introduce cell loss and/or delay. The extent to which these service degradations occur is measured by the quality of service (QOS) offered to the call.

The aim of a preventive CAC scheme is to balance the QOS offered to admitted calls against network utilization by limiting the number of calls using the network. Many schemes described in the literature decide whether or not to accept a call based on knowledge of the *call behavior*, the user's *quality requirements*, and the *current state* of the network. Ideally a user requesting a call would give a complete statistical description of the call, but in practice only a limited indication of expected call behavior is feasible [7]. Call behavior is described by a set of parameters called traffic descriptors, such as mean bit rate, peak bit rate, maximum burst length, probability of cell arrival in a fixed interval, and so on. User quality requirements are usually expressed in terms of acceptable cell loss, delay and jitter. Based on these requirements, traffic sources are divided into classes and each class is provided with a different QOS guarantee, eg. [8]. The current state of the network can be determined by monitoring the utilization of network resources and/or by characterizing the behavior of calls already admitted. For example, traffic models based on fluid flow approximations have been used in analyzing network bandwidth and buffer utilization [9]. Based on the above knowledge, CAC schemes have been developed in which each source is assigned an effective bandwidth [10] in order to meet its QOS while still permitting a statistical multiplexing gain.

There are many problems with these approaches. Users may not know enough about the statistics of their calls to provide accurate traffic descriptors to the network. This is particularly true for certain data transfers with long holding times. For example, in browsing a remote image database, it may not be possible to give a reasonable estimate of the mean bit rate except over a long period of time. Even if users know their traffic descriptors they may be unwilling to reveal them, or may try to misrepresent them. This implies the need for some form of traffic policing and enforcement at the network access points, in order to ensure that each source is complying with what it declared at call setup. Assuming accurate traffic descriptors, a complicated analysis is required to determine whether sufficient resources are available to support the requested call's QOS while not degrading the QOS of admitted calls. Current computational methods cannot do this analysis on the short timescale in which CAC decisions must be made [7]. Consequently either simplifying assumptions about call and resource interactions are introduced, or calls are grouped into classes whose members are assumed to have similar characteristics. Making simplifying assumptions can lead to an over–controlled network, or produce unreliable results. For example, recent studies have shown that LAN and VBR video traffic is self–similar [11], which is very different to the behavior predicted by Poisson–based or fluid flow models. There is a tradeoff between simplicity, which implies a small number of traffic classes, and the large number of classes needed to cover a wide range of different traffics and services. There is likely to be a wide variation in user requirements even within the same traffic class, due to individual preferences and internetworking considerations. Even if accurate traffic and network models were available, technological advances may quickly render their assumptions invalid. For example, the bandwidth required for video or image transfer applications may continue to decrease due to improvements in compression / decompression algorithms and technology [12]. In addition, it is impossible to predict the characteristics of all future applications, or the application mix when users become familiar with new network capabilities. Hence **it may be unwise to base CAC and bandwidth allocation schemes on the requirements and properties of current services.**

The area of CAC and bandwidth allocation is the subject of intensive study and work is in progress addressing the above problems. Reactive control and feedback schemes are also under investigation (eg. [13], [14]) and may be useful in certain network environments or on longer timescales, eg. call level rather than burst or cell level. Experience with real users will show which schemes are feasible in an actual network as opposed to a research or test environment. The problem is complex and is likely to require a multilevel solution approach [15]. It is fair to say that how bandwidth will be allocated in ATM networks is still an open question.

INTEGRATED–SERVICES NETWORKS

The exact form of integrated–services networks of the future is also unclear, but based on current trends in communications and computing it is possible to predict some of its features with reasonable confidence. Multimedia services offering storage and transfer of voice, video, images and data will be supported, allowing remote conferencing and collaboration and replacing text processing with multimedia 'document' processing. Networks will be interconnected and will allow users and organizations to set up virtual private networks embedded in the physical network [16], [17]. Intelligent network interfaces will be needed at internetwork boundaries and at user access points to hide the interface details and present the image of a single network to the users. Users will demand a wide spectrum of application requirements as they become familiar with the network. Some may continue to opt for network–defined services or choose from a library of predetermined services, but others may want to customize their calls and vary call parameters dynamically. Computational power of various types may be offered for rent by the network or third–party providers, enabling users from different fields to access the type of processing appropriate to their needs.

We take the view that current trends and future considerations in the communications and computing areas point to the need for a change in the traditional user–network relationship. The traditional model is centralized with passive users : the network provides well–defined services and users choose from this limited range. User feedback is long–term and inferred from the aggregate demands for services. This is appropriate for large–scale provision of a single service, as in the phone network, but may be unsuitable for the kind of integrated multi–service network described above. The relationship implied by many of the proposed CAC schemes is more of a 'contract' : users describe their traffic and make quality demands, and the network provides a stated level of service while enforcing the user commitments. Some problems with this model were outlined above, although it is a step in the right direction. Taking this process further, our view is that

- the network should be a provider of **resources** rather than services;
- the responsibility for packaging these resources into services should lie with the **users** (or in real–time their interface equipment);
- the network's primary function should be to **coordinate requests** for its resources. The goal of this coordination could be to optimize some measure of network performance, or to maximize a suitably–defined global user satisfaction, or to ensure some degree of fairness, or some other objective.

Thus we view the network simply as a high–speed cell relay network in which the central issue is transporting cells rather than implementing services defined at higher levels [4]. Our view is consistent with that expressed in [18] that an ATM virtual circuit is similar to a virtual wire, since the service delivered by the virtual circuit is similar to that of the physical layer in the OSI reference model. Users may actually be working with higher layer protocols such as TCP/IP running on top of an ATM network, which also argues for keeping the ATM network as simple as possible. One proposal along these lines is to move call admission and bandwidth allocation decisions to the terminal equipment at the network access points, and somehow ensure that the combined user rates do not exceed the network capacities [19]. In our view, users take responsibility for requesting sufficient network resources to meet their QOS requirements. At call setup this involves specifying some parameters of the requested connection, either manually or in a menu–driven environment. If users want to adjust connection parameters during a call then these parameter specifications may be automated. This kind of in–call negotiation is desirable for bursty calls where the actual resource usage is of interest [20]. One traffic management scheme for burst–level resource reservation is described in [7], and a form of in–call negotiation is now commercially available in some network interface equipment. Even for static connection requests, the call setup process may involve a negotiation between the user and the network in which the user modifies their request to conform to the current level of resource utilization. For example [21] outlines a real–time channel setup procedure in which detailed information is sent to the user if a connection request is refused, allowing the user to take this feedback into account in a revised request. Some work is underway on modifying user traffic inputs via pricing in the Internet [22] and in integrated–services networks [23], [24]. A contract–based CAC scheme is proposed in [25] in which the charge to users is related to how accurately they declared their traffic, rewarding users who provide better information on their call characteristics.

This view of the network may require a new approach to the CAC and bandwidth allocation problems. We have developed one solution approach which is compatible with this new user–network relationship. Our approach is based on users regulating their traffic inputs in response to bandwidth prices calculated by the network. The network and users are considered to form a system and we apply economic principles of pricing for resource allocation to this system. Another traffic management scheme similar in spirit to our approach is discussed in [26].

PRICING BANDWIDTH

The system made up of the users together with the network has various resources that can be used to meet service demands. However in all realistic systems these resources are limited and some method of allocating them is needed when total demand is greater than the resource limit. In this paper the resources are the capacities of the ATM connections. The bandwidth allocated to a user is considered to be a commodity which is sold by the network to the user [27], [28]. We view the users as placing a benefit, or willingness–to–pay, on the bandwidth they are allocated. Given a price per unit of bandwidth, a user's benefit function completely determines that user's traffic input. Users are assumed to act in their own best interests and to be capable of responding to changes in the price for bandwidth.

In our formulation, network constraints such as virtual path capacities [29] are translated into cost functions on the bandwidth demands. This is because one user's consumption of bandwidth gives rise to a social 'cost' (in terms of longer delays, less available bandwidth, etc) which is borne by all users. The network operator sets the prices so that the marginal benefit the users place on their bandwidths is equal to the marginal cost of handling that traffic in the network[1]. The network operator dynamically adjusts the prices based on monitored network conditions. We assume that when the price of bandwidth increases, users reduce their demands. There is no need for enforcement of call parameters in our scheme because users do not declare anything about their traffic. The mathematical formulation of the scheme is in an Appendix.

A typical configuration that we envisage is shown in Figure 1. The user programs the terminal equipment with their benefit function. The bandwidth in cells per second allocated to the user by the network depends on this benefit function and the price that is sent from the network to the terminal equipment.

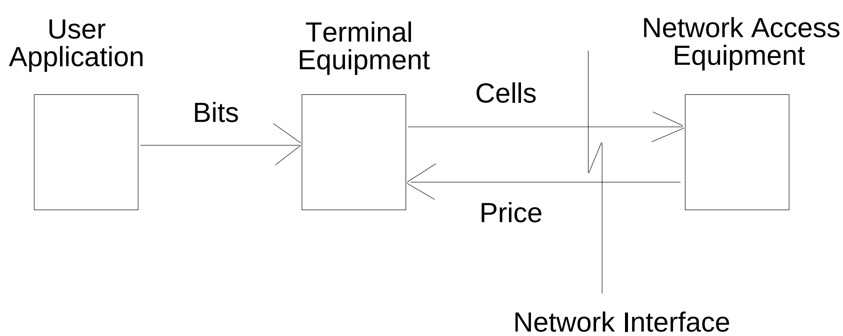

Figure 1. Proposed configuration for access

Our approach is intended to be applicable to general ATM networks and to the full range of user and service types, such as voice, video, real–time data and non–time–sensitive data. However a special case that is particularly suitable is that of a Virtual Private Network (VPN) in which one operator controls the network and all applications running on it. The relative importance of various applications could then be varied over time and from node to node. For example, the operator may place a high priority on operational data transfer and low priority on other data transfers; or interactive video and voice might get higher priority than other traffic types.

However, assigning dynamic priorities is difficult. If the real–time applications such as voice and video are given priority to ensure timely delivery, then data traffic may suffer higher loss though it may not be able to tolerate cell loss as well as voice. On the other hand, if priority is given to data and large buffering is employed, then real–time applications may suffer large variable delays.

What is needed is a dynamic adaptive intertemporal priority scheme. The priorities should change to track changes in the network state or in the application requirements over multiple time periods. Rather than have a complicated priority scheme, a pricing scheme like the one proposed in this paper could be used. The operator would set the benefit functions for the different applications, and could also set different benefit functions for

[1]We address only variable costs corresponding to network constraints. The recovery of fixed network costs could be done by an access charge or some other method.

applications of the same type. Each application would then input traffic according to its assigned benefit function and the current state of the network, as reflected in the prices. We have simulated a VPN and implemented a simplified pricing scheme to investigate this possibility. In the rest of the paper we describe our simulation model and the results we have obtained with it so far.

2. SIMULATION MODEL

To investigate our proposed algorithm and its efficiency, we built a model of a single link which represents a single virtual path. Typically there could be diverse services using this path and so a number of models of sources were needed. We implemented three basic source models, representing some of the services that will be demanded in an ATM network : voice, video and data. For simulation purposes we investigated a link with a bit rate of 2.048 Mbps, although our results are expected to scale to higher link speeds.

The voice model that we use is a standard model with two states, speaking and silence [30], [31]. During speaking periods, cells are produced by the source at the bit rate of the voice. During silent periods no cells are produced. The mean duration of the speaking (or ON state) is 0.352 seconds. If we count the number of cells produced in this period it is found that this number is geometrically distributed. In our case this geometric distribution will have a mean of 59 cells. The silence is made up of two parts, one from the silences when listening and the other from the silences between words or sentences. Therefore to properly represent the silence distribution there should be two distributions. However it has been found that a hyper–exponential distribution will accurately model the silences, with the lower exponential for the inter–word silences and the upper exponential for the listening silences. The average silence is 0.65 seconds long. The probability of being in the lower exponential (rather than the upper exponential) is 0.8746. From these parameters it is possible to fully characterize the hyper–exponential silence distribution. The two means of the silence distribution are 0.372 seconds and 2.592 seconds. The standard deviation of the distribution is then 1.22525. The resulting model assumes an input of 64 kbps voice which if placed directly in the ATM layer would require a bit rate of 71 kbps due to the 5 byte overhead for every 48 bytes of data. However as only the speaking portion is sent the required bit rate is 24.8 kbps. This gives a peak–to–mean ratio of 2.8. The model outlined here assumes 64 kbps voice, but this can be dynamically changed during the simulation if desired.

One of the major intended uses of ATM networks is for video applications. The video source model that we use here is a standard one for video conferencing [32]. The codec is a one layer model with a compressed bit rate of 222 kbps, which adheres to the H.261 standard for video telephony. The model used is a discrete markov model with nine states. These states capture the transitions from one average bit rate to another. The maximum bit rate delivered by the codec is 750 kbps. Within a state the bit rate is assumed to be uniformly distributed. The transition probability was engineered so that the matrix only had diagonal and one–off–diagonal entries. This allows a birth–death queueing structure to be formed. Another possible model is based on two layer codecs. In this model the base layer is approximately CBR and if this bit rate can be reserved the base layer delivers a minimum level of quality. The enhancement layer is VBR with a peak–to–mean ratio of maybe 40, and is used to enhance the picture. The advantage of using a two layer codec is that the enhancement layer data can be treated as best effort traffic. However it is unlikely that the same compression can be achieved with two layer codecs as with one layer codecs.

A data source is one of the most difficult sources to model as the source type depends on what applications are being run and on what systems. In this paper we model a file transfer application. This captures the bursty nature of data communications as well as its looser delay requirements relative to voice and video. A model was built based on transferring files from one computer to another. An empirical distribution for file size ranges was obtained from actual files stored on one of our computers. In the simulations a range was chosen according to this empirical distribution, and then a file size was chosen from a uniform distribution within this range. The amount of data transfer can be varied depending on how many files are to be transferred. For one file per second the average bit rate is 197 kbps. The peak–to–mean ratio of this source can be high with values up around 1000. The user's quality requirement is that the file will be transferred at a rate determined by the network. A rate is given to the data source for a time interval and the rest of the file is stored awaiting transfer.

Simulation Environment

The network and source models were simulated using SES/workbench [33], a discrete-event simulator that allows hardware and software simulation. The models in this paper were mainly created by use of its graphical user interface. SES/workbench compiles the graphical code to C and creates an executable. The simulation execution platform was a cluster of Sparc-10 workstations. The run time for the simulations was 1000 seconds, or 5 million cells. Sharing the 2.048 Mbps link were 20 voice sources, 2 video sources and a data source. A schematic of the simulation setup is shown in figure 2 below.

The simulation model is made up of submodules, each of which performs a well defined function. The sources generate cells which are input to a network interface submodule. The network interface takes the source bit stream and forms ATM cells. For voice cells 5 bytes of header are added to the 48 byte information load. For data and video cells we implemented AAL 3/4 which uses 44 bytes of information with a 4 byte AAL header and trailer in addition to the 5 byte ATM header. The cell stream from an interface is then input to the ATM switch buffer submodule. This submodule smooths the arrival of cells to the ATM network and so takes care of cell scale congestion. This buffer is the limited resource critical to the operation of the model. In a typical implementation the buffer is managed by an input control mechanism, such as a leaky bucket scheme.

The leaky bucket is a method for policing the mean rate of a source and still allowing bursts to occur. The mean rate is the rate at which tokens are given to allow cells past the network interface and into the network. If there were no storage of tokens then this would be a pure mean rate policing function. If the source can store tokens, it could burst all those tokens in one go. If a source has no remaining token allowance then rather than dropping that cell usually it is merely marked. If there is congestion in the network and cells have to be discarded the marked ones are disacarded first. The leaky bucket does not lower the cell loss in a buffer; it merely selects which ones to lose. Our pricing scheme

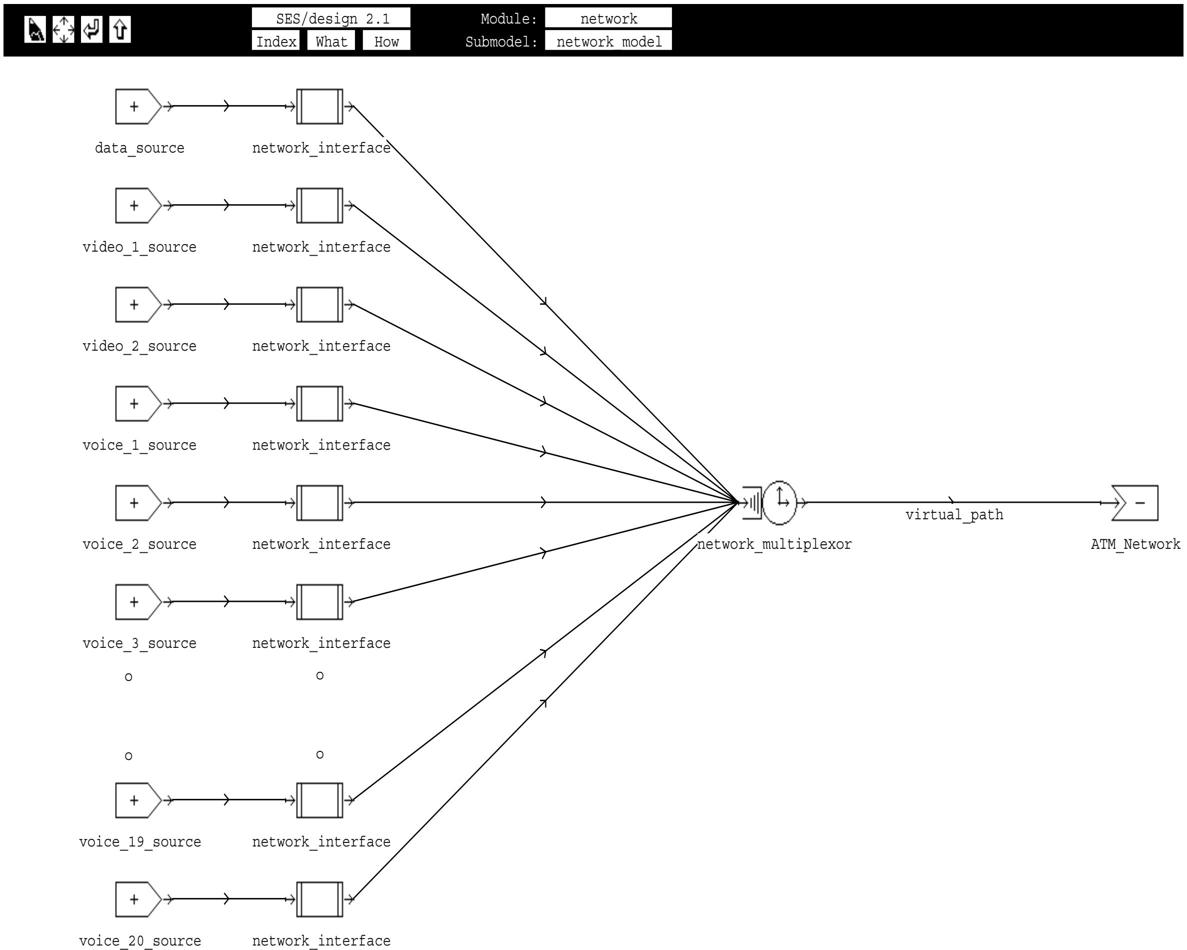

Figure 2. Simulation Schematic

actually lowers the probability of cell loss by smoothing the traffic adaptively based on the buffer occupancy to avoid congestion.

In our proposed scheme a price is generated by the network based on the present state of the network buffer, and the sources adapt their demands based on this. The leaky bucket can also be implemented on top of our scheme so that if there is cell loss we can discard the marked ones first. The model takes in cells over an interval of about 50 msec and gives a price to all the sources sharing the link. The price reflects the congestion in the buffer (if any) and hence on the virtual path. The higher the buffer occupancy, the higher the price sent back to the sources. This price is received by the network interfaces which use it to calculate how many cells they *should* input in the next interval. For example, a data source may calculate that at the given price it should input 100 cells to the network, but if no cells are actually generated by the data source in that time interval then obviously there is no input. The outline of the distributed pricing algorithm we used is contained in the Appendix.

In an interval of about 50 msec the link can serve about 250 cells and the buffer size we choose is 250 cells. This means that when the buffer is full there should be no more than 250 input cells over the next interval. The input sources are shaped as much as possible by the time the cells are formed at the network interfaces, so the cells arrive approximately equally spaced in time. It is standard to guarantee real–time services that if their cells get through then the maximum access delay they experience is the buffer queueing delay which in this case is small (50 msec).

3. RESULTS

In practice voice and video calls may be inflexible with respect to bandwidth, and even if they are flexible they do not permit a huge statistical multiplexing gain (except for the enhancement layer of a two layer codec) compared to data. In our simulations we assumed that the voice and video have inflexible bandwidth demands. We used 20 voice sources which have a combined mean of 492 kbps and a potential maximum of 1413 kbps. The video has two channels, both one layer codecs, with a combined mean of 507 kbps and a potential maximum of 1773 kbps. Even without a data source there is clearly a nonzero probability of cell loss. The data source has a mean of 237 kbps and a potential maximum equal to the bit rate of the link, 2048 kbps.

When a leaky bucket scheme alone was implemented the results in Figure 3 were obtained for the sampled buffer occupancy. From an analysis of the results, there were 32,268 marked cells and 191,044 unmarked cells lost out of a total of 2,914,191 generated over the 1000 seconds. This gives a loss probability for the marked cells of 1.1 percent and for the unmarked cells of 6.6 percent.

In Figure 4 we show the plot of buffer occupancy over the interval when our pricing scheme was used.

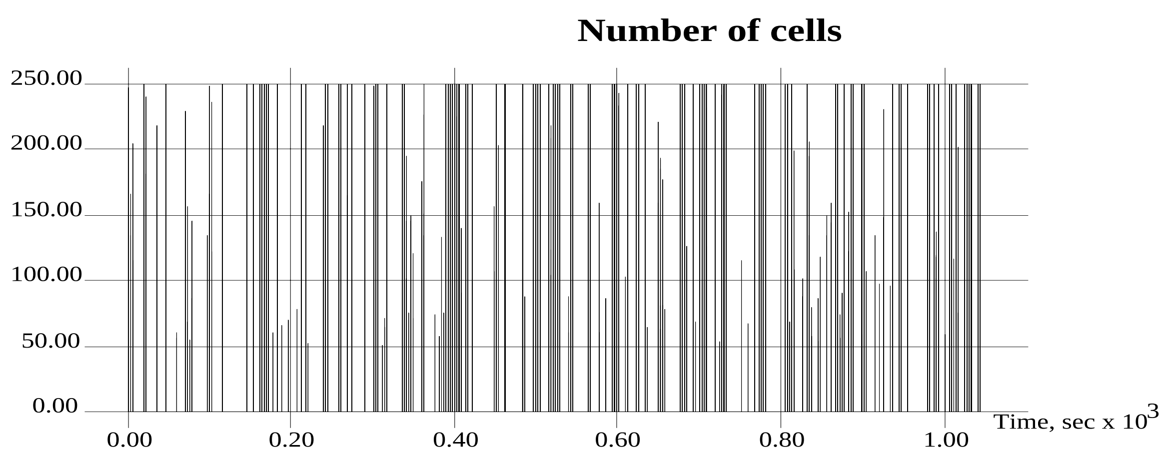

Figure 3. Buffer Occupancy with Leaky Bucket.

The buffer is not as occupied as in Figure 3. This is apparent when we compare the loss probabilities. The pricing scheme produced no cell loss, either of marked or unmarked

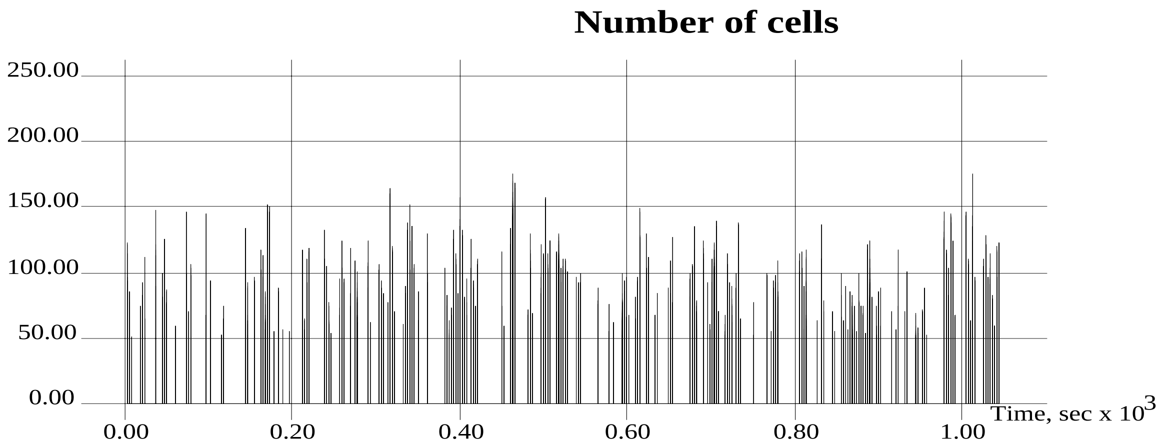

Figure 4. Buffer Occupancy with Pricing.

cells[2]. There is a small decrease in the quality of service provided to the data source to gain this improvement. Figure 5 shows the sampled number of waiting data cells of data over the interval when the leaky bucket is used. By comparing Figure 5 to Figure 6 for the pricing scheme, we see that the mean waiting time for the data source has increased.

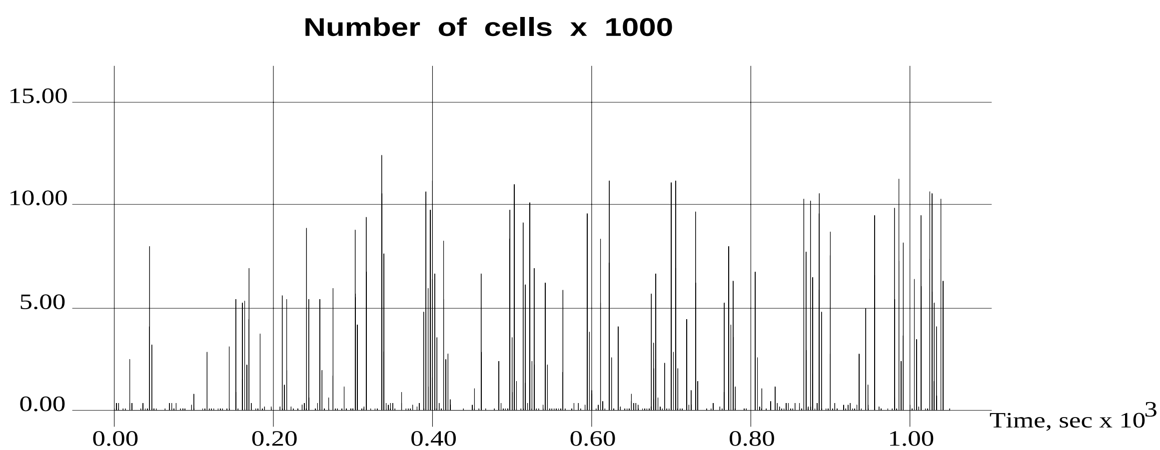

Figure 5. Waiting Times for Data Cells with Leaky Bucket.

4. DISCUSSION AND CONCLUSIONS

This paper suggests pricing and user self–regulation as a means of allocating bandwidth in ATM networks. The pricing scheme is independent of the traffic arrival model and does

[2]It should be noted that there could be cell loss if a longer interval was chosen.

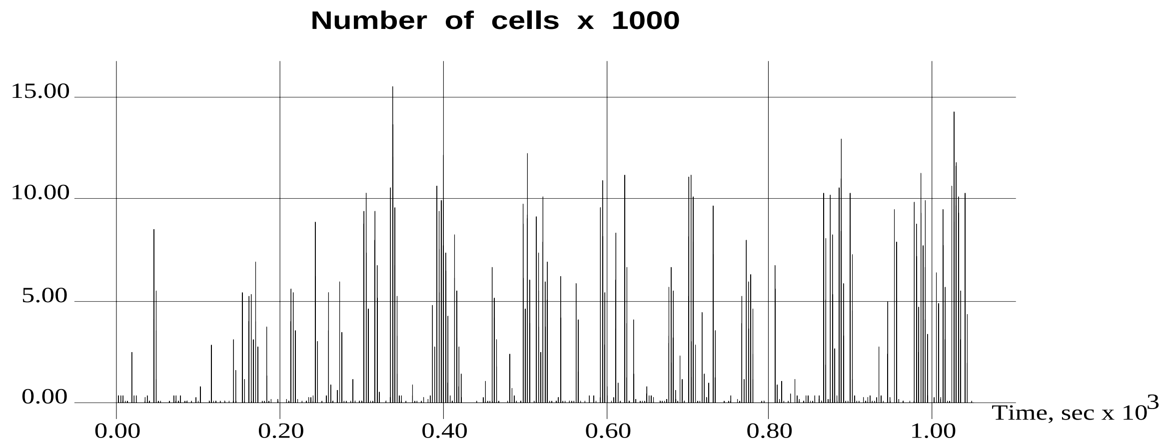

Figure 6. Waiting Times for Data Cells with Pricing.

not require the network to police and enforce user commitments in real–time as cells arrive from the sources. The pricing algorithm is distributed and local to the ATM access nodes, suggesting that it may be able to meet the tight time requirements of an admission control scheme. There is no need for the network to define and support traffic classes. The users have more flexible access to network resources than in other CAC schemes, but they take the responsibility of determining how much network resources to acquire in order to meet their particular needs. It should be noted that this view of the user–network relationship should not be judged based on the feasibility or otherwise of our pricing scheme. Other CAC schemes may also be suitable and it remains to be seen whether our scheme is compatible with other approaches, perhaps in a multi–level solution approach. Similarly our pricing scheme offers users the possibility of taking more control over their service than they have under traditional system models, but it does not *require* them to. Users whose applications have inflexible bandwidth demands, or who set a threshold price above which they demand zero bandwidth, can be accommodated in our formulation. In order to pursue an inflexible strategy, users may have to pay premium prices at times of heavy demand; but it is our belief that simply being forced to examine the relative importance of their usage will give rise to a variation in user valuations that a scheme like ours can exploit. The success of our approach depends on having a sufficient number and range of flexible applications, although how much is enough is as yet unknown.

In VPNs pricing can be used to choose the desired application mix and vary this mix dynamically, as demonstrated in a simple case by our results. Another use of pricing is in selectively dropping cells during congestion. Suppose that an application uses Ethernet frames and one cell is dropped from a frame. Then the benefit of the rest of the frame should be set to zero if the whole frame has to be retransmitted. Similarly, if there are a number of video calls in progress, it may be better to drop all the cells from one call during congestion rather than to spread out the loss. The feasibility of using our pricing approach as a dynamic adaptive priority scheme in this way is one avenue for further research.

Our pricing scheme also applies to general ATM networks in which the user benefit functions are unknown to the network operator. There are many issues which would have to be resolved in a practical implementation. For example, our assumption here is that the benefit of making a call is associated entirely with the sender. But in practice the benefit is shared, and some pre-call negotiation may be necessary to agree on call parameters acceptable to all parties. If intermediate ATM nodes switch on the VCI as well as on the VPI [1], users at different ATM nodes have to compete for shared resources, which complicates the algorithm and may increase the minimum length of a pricing interval. Whether the prices used in our scheme could be used as part of an ATM network tariff is not clear. However we believe that the paradigm suggested in this paper is an interesting one and may be helpful in solving these difficult network management problems.

Acknowledgements

We are grateful for the help and advice given by the late Ed Posner of Caltech and JPL in the initial stages of this work. We thank Pravin Varaiya, Gustavo de Veciana, Mike Wong, Padmanabhan Srinagesh, Mike Honig, Steve Low and Jerry Teahan for helpful discussions about this work. The first author acknowledges Charles McCorkell of Dublin City University for his continuing support.

APPENDIX

We briefly describe the mathematical formulation of our bandwidth allocation algorithm here. A fuller treatment can be found in [34].

We assume that a virtual path is set up for every source–destination pair, and all virtual circuits using a virtual path are for the same source–destination pair. Thus when a call arrives at the access to the network it is assigned a virtual circuit, which is amalgamated on one virtual path with the virtual circuits for other calls from that source to destination, as shown in Figure 7.

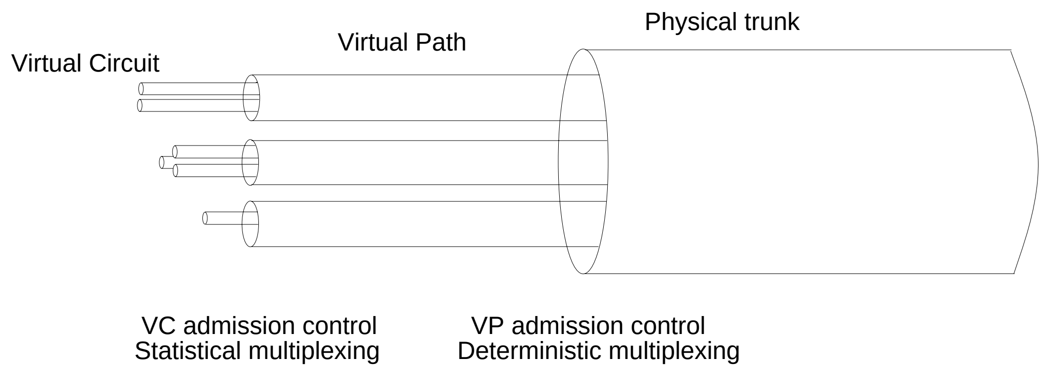

Figure 7. Virtual Path/Virtual Circuit connections

The virtual paths are carried by the physical network trunks, whose capacities are assumed to be fixed. We assume that statistical multiplexing of the sources is done in the sense of allocating a capacity to a virtual path which is less than the sum of the virtual circuit peak bandwidths using this virtual path. The network decides the capacities of the virtual paths and these satisfy the physical trunk capacity constraints. For example, the network could simply do deterministic multiplexing at the virtual path level, as in Figure 7. In this paper the virtual path capacities are assumed to be fixed, although this assumption can be relaxed [34].

The users connected to an ATM switch wish to send traffic to various destinations. Their benefit versus bandwidth curve for a particular call is assumed to be concave increasing in general, as shown in Figure 8. This follows the usual economic assumption that users value the first part of a commodity the most, with diminishing incremental valuation as they get more of the commodity [27],[35].

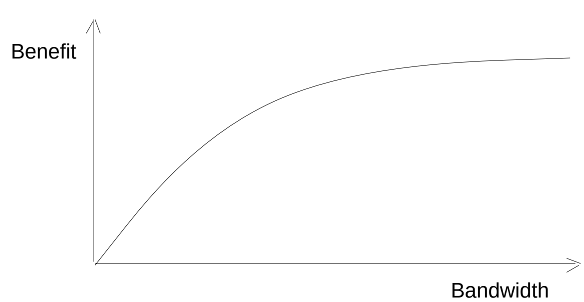

Figure 8. User Benefit as a function of Traffic Rate

Consider the problem faced by a user attached to ATM node i in deciding their bandwidth demand for a particular call to destination ATM node j. Let the virtual path corresponding to this source–destination pair be denoted V_q. The network quotes all users a price per unit of bandwidth on V_q, π_q. Each user then uses their private benefit function ben_{rq} for call r to decide their bandwidth at that price, b_{rq}. This decision process can be formulated mathematically as follows :

$$maximize \quad ben_{rq}(b_{rq}) \; - \; \pi_q \cdot b_{rq}$$

The user solves a maximization problem of this form for each call to destination ATM node j, for each j. The optimality condition for this maximization is [36]

$$\frac{\partial ben_{rq}}{\partial b_{rq}} \; - \; \pi_q \;\; = \;\; 0 \, , \quad \forall \;\; V_q \tag{1}$$

Consider now the system made up of all the users and the ATM network, and suppose for the moment that there is a single 'system manager' who knows all the problem data. We will see later on that no such 'system manager' is needed.

The problem facing this (hypothetical) system manager is to choose the bandwidth demands to maximize aggregate benefit, subject to the virtual path capacity constraints. Here we replace the virtual path capacity constraint by a corresponding buffer capacity constraint, which ensures that the ATM switch buffer occupancy b_q is less than the buffer size B_q but allows the input rates to temporarily exceed the output rate of the virtual path :

$$b_q \;\; \leq \;\; B_q \; , \quad \forall \;\; V_q$$

We drop this constraint but add a barrier function term to the system objective function corresponding to it, $Buffcost_q$. For example,

$$Buffcost_q \; := \; \frac{1}{B_q \; - \; b_q} \; - \; \frac{1}{B_q}$$

The shape of this cost term is shown in Figure 9, and ensures that as the buffer capacity is approached, the 'cost' of assigning bandwidths increases to infinity.

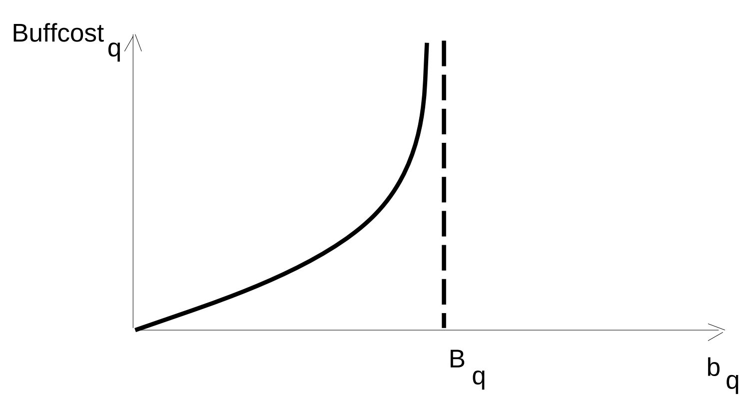

Figure 9. Barrier Function For q^{th} Buffer Capacity Constraint

The system problem can be formulated as follows :

$$maximize \quad \{ \sum_{all\ V_q} [\sum_r ben_{rq}(b_{rq})] \ - \ Buffcost_q \}$$

The optimality conditions for the system problem are [36]

$$\frac{\partial ben_{rq}}{\partial b_{rq}} \ - \ \frac{\partial Buffcost_q}{\partial b_q} \cdot \frac{\partial b_q}{\partial(\sum_r b_{rq})} \ = \ 0 \ , \quad \forall\, r, V_q \tag{2}$$

Comparison of equations 1 and 2 shows that by setting

$$\pi_q \ = \ \frac{1}{(B_q \ - \ b_q)^2} \cdot \frac{\partial b_q}{\partial(\sum_r b_{rq})} \tag{3}$$

individual user benefit maximization (equation 1) coincides with system optimality (equation 2). Therefore if the network operator sets the prices according to equation 3, the users will self-select their optimal bandwidth demands while competing for the virtual path capacity. There is no need to have a 'system manager' who knows all the user benefit functions as well as the network data. The network operator can induce system-optimal user behavior by setting the prices as in equation 3 without any knowledge of the user benefit functions.

Distributed Pricing Algorithm

Let time be divided into successive pricing intervals of length T. Within each interval the benefit functions of all the users are assumed to be fixed; however, these benefit functions may change from one interval to the next. The price per unit of bandwidth on virtual path V_q, π_q, is announced by the network at the start of each pricing interval, and remains fixed for the duration of the interval. Hence each user solves a maximization problem to decide their optimal average bandwidth b_{rq}, and transmits exactly $b_{rq} \cdot T$ cells during the interval. At each ATM node, the resulting buffer occupancies are measured, enabling the network operator to calculate the marginal cost of each buffer with respect to its total traffic input rate, as given by the right-hand side of equation 3.

If these marginal costs are equal to the prices that were announced, then the system optimality conditions in equation 2 are satisfied. If not, then the prices must be updated to correct for the difference. In this simulation study we used an update rule based on the method of False Position.

We summarize this negotiation between the network and the users by the following Bandwidth Allocation Algorithm :

Step 0. network operator chooses initial values for the prices $\{\pi_q\}$

Step 1. network operator announces these prices to the users at the start of the current pricing interval

Step 2. users respond by choosing their $\{b_{rq}\}$ according to equation 1

Step 3. network operator calculates the marginal costs of the buffers with respect to the bandwidths, as on the right-hand side of equation 3

Step 4. network operator adjusts the prices $\{\pi_q\}$ to reduce the difference between the terms in equation 3$\Rightarrow$go to Step 1

Note that the Bandwidth Allocation Algorithm is purely local to an ATM node and so can be done at the ATM node to which the users are attached. This is because we have assumed above that the virtual path capacities are fixed.

REFERENCES

1. M. de Prycker, *Asynchronous Transfer Mode : Solution for Broadband ISDN*, Ellis Horwood, 1991.
2. P. Komisarczuk, M. H. Hadjitheodosiou, F. Coakley and C. Smythe, 'B-ISDN Implementation Via Satellite : The Catalyst Project', *10th UK Teletraffic Symp.*, BT Laboratories, Martlesham Heath, April 1993.
3. J. Murphy, E. Chow and R. Markley, 'ATM Service-Based Selective Retransmission Over DSN Satellite Links', to be presented at the 15th AIAA International Communication Satellite Systems Conference, San Diego, USA, March 1994.
4. I. M. Leslie, D. R. McAuley and D. L. Tennenhouse, 'ATM Everywhere ?', *IEEE Network*, Vol. 7, No. 2, pp 40-46, March 1993.
5. I. Nikolaidis and R. O. Onvural, 'A Bibliography on Performance Issues in ATM Networks' *Computer Communication Review*, Vol. 22, No. 5, pp 8-23, Oct. 1992.
6. V. E. Benes, 'Programming and control problems arising from optimal routing in telephone networks', *Bell Syst. Tech. J.*, Vol. 45, pp 1373-1438, 1966.
7. J. S. Turner, 'Managing Bandwidth in ATM Networks with Bursty Traffic', *IEEE Network*, Vol. 6, No. 5, pp 50-58, Sept. 1992.
8. L. Trajkovic and S. J. Golestani, 'Congestion Control for Multimedia Services', *IEEE Network*, Vol. 6, No. 5, pp 20-26, Sept. 1992.
9. S. Low and P. Varaiya, 'A Simple Theory of Traffic and Resource Allocation in ATM', *Proc. of Globecom '91*, pp 1633-1637, Dec. 1991.
10. F. P. Kelly, 'Effective Bandwidths at Multi-Class Queues', *Queueing Systems Theory and Applications*, Vol. 9, No. 1-2, pp 5-15, Oct. 1991.
11. W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, 'On the Self-Similar Nature of Ethernet Traffic', *Computer Communication Review*, Vol. 23, No. 4, pp 183-193, Oct. 1993.
12. G. J. Armitage and K. M. Adams, 'Using the Common LAN to Introduce ATM Connectivity', *IEEE 18th Conf. on Local Computer Networks*, Sept. 1993.
13. H. Kanakia, P. P. Mishra and A. Reibman, 'An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport', *Computer Communication Review*, Vol. 23, No. 4, pp 20-31, Oct. 1993.
14. H. Gilbert, O. Aboul-Magd and V. Phung, 'Developing a Cohesive Traffic Management Strategy for ATM Networks', *IEEE Commun. Mag.*, Vol. 29, No. 10, pp 36-45, Oct. 1991.
15. S. Yazid and H. T. Mouftah, 'Congestion Control Methods for BISDN', *IEEE Commun. Mag.*, Vol. 30, No. 7, pp 42-47, July 1992.
16. R. Ahmad, F. Halsall, K. Lam and J. Zhang, 'Performance Evaluation of Private ATM Networks', *10th UK Teletraffic Symp.*, BT Laboratories, Martlesham Heath, April 1993.
17. J. M. Schneider. T. Preuss and P. S. Nielsen, 'Management of Virtual Private Networks for Integrated Broadband Communication', *Computer Communication Review*, Vol. 23, No. 4, pp 224-237, Oct. 1993.
18. M. de Prycker, R. Peschi and T. van Landegem, 'B-ISDN and the OSI Protocol Reference Model', *IEEE Network*, Vol. 7, No. 2, pp 10-18, March 1993.
19. S. M. Walters, 'A New Direction for Broadband ISDN', *IEEE Commun. Mag.*, Vol. 29, No. 9, pp 39-42, Sept. 1991.
20. J. W. Roberts, 'Variable-Bit-Rate Traffic Control in B-ISDN', *IEEE Commun. Mag.*, Vol. 29, No. 9, pp 50-56, Sept. 1991.
21. D. Ferrari, J. Ramaekers and G. Ventre, 'Client-Network Interactions in Quality of Service Communication Environments', *IFIP Transactions C (Communication Systems)*, Vol. C-14, pp 221-234, 1993.

22. J. K. MacKie-Mason and H. R. Varian, 'Pricing the Internet', presented at *Public Access to the Internet*, JFK School of Government, Harvard University, May 26-27, 1993.
23. S. Shenker, 'Service Models and Pricing Policies for an Integrated Services Internet', presented at *Public Access to the Internet*, JFK School of Government, Harvard University, May 26-27, 1993.
24. C. Parris, S. Keshav and D. Ferrari, 'A Framework for the Study of Pricing in Integrated Networks', Technical Report TR-92-016, International Computer Science Institute, Berkeley, CA, March 1992.
25. F. P. Kelly, 'Tariffs, Policing and Admission Control for Multiservice Networks', *10th UK Teletraffic Symp.*, BT Laboratories, Martlesham Heath, April 1993.
26. S. H. Low and P. P. Varaiya, 'A New Approach to Service Provisioning in ATM Networks', to appear in *IEEE Trans. on Networking.*
27. A. de Fontenay, M.H. Shugard and D. S. Sibley, *Telecommunications Demand Modeling : An Integrated View*, North-Holland, 1990.
28. J. R. Meyer, R. W. Wilson, M. A. Baughcum, E. Burton and L. Caouette, *The Economics of Competition in the Telecommunications Industry*, Oelgeschlager, Gunn and Hain, 1980.
29. J. Burgin and D. Dorman, 'Broadband ISDN Resource Management : The Role Of Virtual Paths', *IEEE Commun. Mag.*, Vol. 29, No. 9, pp 44-48, Sept. 1991.
30. J. G. Gruber, 'A Comparison of Measured and Calculated Speech Temporal Parameters Relevant to Speech Activity Detection', *IEEE Trans. Communs.*, Vol. COM-30, No. 4, pp 728-738, 1982.
31. P. T. Brady, 'A Statistical Analysis of On-Off Patterns in 16 Conversations', *Bell Syst. Tech. J.*, pp 73-91, 1968.
32. 'Redundancy Reduction Techniques for Coding of Video Signals in Multi-Media Services', COST 211ter, Compendium ATM, Coding Procedures, Simulation Subgroup, pp 1-83, May 1993.
33. SES, *SES/workbench Reference Manual*, Release 2.1, Feb. 1992.
34. J. Murphy, L. Murphy and E. C. Posner, 'Distributed Pricing For Embedded ATM Networks', to be presented at *International Teletraffic Congress ITC-14*, Antibes, France, June 1994.
35. US Department of Commerce, *Telecommunications in the Age of Information*, Oct. 1991.
36. D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd Ed., Addison-Wesley, 1984.