

Distributed Pricing For Embedded ATM Networks

John Murphy*, Liam Murphy[†] and Edward C. Posner¹

*Electronics Engineering, Dublin City University, Ireland².

[†]EECS Department, University of California, Berkeley, USA.

This paper addresses the problem of bandwidth allocation in ATM networks. Users are assumed to place a benefit on the bandwidth they are assigned by the network. The network constraints are used to derive prices for the bandwidths which induce users to share the limited network resources according to their benefit functions. This formulation leads to a distributed pricing algorithm, which is integrated with the problem of assigning capacities on the virtual paths connecting ATM switches such that the physical trunk capacity constraints are satisfied. The algorithm can be used for traffic management in a Virtual Private Network (VPN) in which one entity controls all the users.

1 INTRODUCTION

The emerging Asynchronous Transfer Mode (ATM) technology is based on the switching of small fixed length packets [1]. The motivation behind ATM is that the network should be able to support the increasing number of diverse digital services that users now require, or may require in the future. The problem we address in this paper is how to support different types of service demanded from an ATM network.

Previous telecommunications networks were designed with one service in mind [2], so the performance of those networks was optimized for that service. With an increasing number of digital telecommunications services being offered, this traditional approach requires many networks to be designed, installed, maintained and operated. Furthermore, when a new service becomes available, this approach suggests a completely new network should be designed to optimize its performance. However it is difficult to predict the performance requirements for new services. This difficulty, and the inability to share existing resources across different networks, leads to inefficiencies.

An alternative approach is to design one network that is able to accommodate all services that exist now or may exist in the future. ATM is an example of this approach. In ATM the available resources are shared between all services, such that an optimum statistical sharing of those resources is obtained [3]. As new services have unknown requirements, it may be impractical to base bandwidth allocation schemes on the requirements of present services. One proposal is to move these decisions to the terminal equipment at

¹Dr. Posner was with Jet Propulsion Laboratory and California Institute of Technology

²This work was done at the California Institute of Technology

the edge of the network, and somehow ensure that the combined user rates do not exceed the network capacities [4].

When a call arrives at the network the network must decide whether to admit that call, if the necessary resources are available, or else reject it. One possibility is to allocate the peak bandwidth to the call. However this does not take advantage of statistical multiplexing of the sources. Another possibility is to allocate less than the peak rate, at the expense of cell loss and delay. In such a scheme it is necessary to know the statistics of all the sources in order to decide on call admission and bandwidth allocation. In addition, for the network to find out what resources are needed, the source must inform the network of the call statistics. The source may not have any incentive to do this, or may not even be aware of what the characteristics will be [5], as in a data call. Even if this information were available, the network must police the call to make sure that it is adhering to what was specified at call setup. Many methods have been proposed to deal with admission and policing for bandwidth allocation. While policing the peak rate is possible, trying to enforce the mean rate is difficult. Kelly notes that ‘many believe that mean rate policing is impractical’ [5].

The bandwidth allocation method we propose in this paper is to have users self-regulate their traffic demands. The self-regulation mechanism is based on economic principles of pricing for resource allocation. In this case the resources are the capacities of the virtual paths. The bandwidth given to a user is considered to be a commodity which is sold by the network to the user [6], [7]. We view the users as placing a benefit, or willingness-to-pay, on the bandwidth they are given. Given a price per unit of bandwidth, a user’s benefit function completely determines the user’s traffic input. Users are assumed to act in their own best interests, independently of all the other users, and are capable of responding to changes in the price for bandwidth.

In our formulation, network constraints such as virtual path capacities are translated into cost functions on the bandwidth demands. 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 network³. The prices are adjusted dynamically based on monitored network conditions. It is assumed that when the price of bandwidth increases, users reduce their demands. There is no need for enforcement of call parameters in our scheme, because there is no pre-agreed contract between users and the network. A typical configuration that we envisage is shown in Figure 1.

This kind of self-regulation of traffic sources is technologically feasible, though for short pricing intervals the user responses would have to be automated. Examples of self-regulation can be given for voice, video and data. In the case of a voice call, suppose the network is not busy so that the price for bandwidth is low and the source is demanding 64 kb/s or more to get high quality voice. As the network gets congested and the price increases, the source may reduce its bandwidth demand to 32 kb/s by using adaptive PCM. As the price increases further, the source may further reduce its bandwidth by using more complicated coding schemes, like eliminating the silences in the speech or giving less bits per sample. This will degrade the quality of the speech but in a graceful way. An example of a benefit function that a user could place on a voice call is shown in

³We address only variable costs corresponding to network constraints, and do not discuss the recovery of fixed network costs.

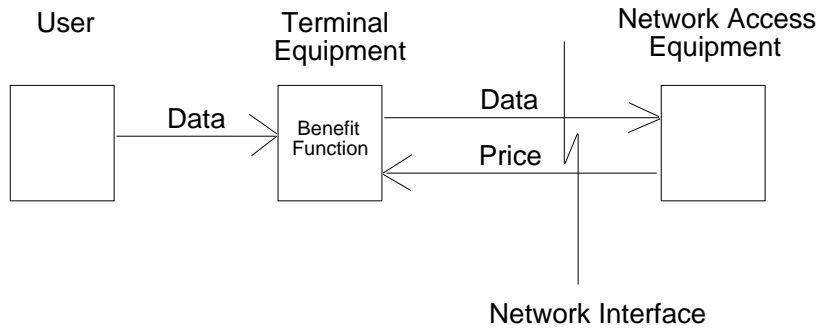


Figure 1: Proposed configuration for access

Figure 2. We approximate this as a smooth function so that we can formulate a continuous optimization problem. Technological improvements may make this approximation closer to an actual benefit function. For video calls a similar argument can be made and the quality of the video degrades as the bandwidth reduces. For data calls, such as file transfer, the degradation is in response time.

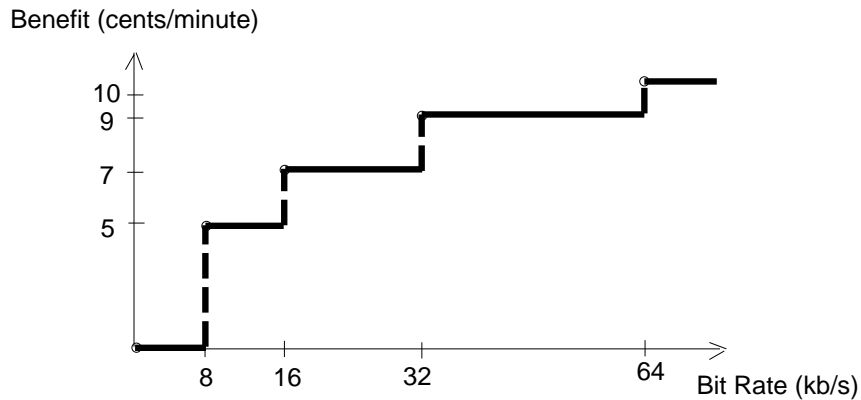


Figure 2: Typical benefit function for a voice call

The users would not continually negotiate for bandwidth manually but rather by means of an intelligent network interface, as in Figure 1. This interface can be programmed by the user with their benefit function and their requirements for each type of service. Thus access is service dependent but the network itself is not, and hence the network is future-proof.

2 SYSTEM MODEL

We model the ATM network as consisting of virtual paths and virtual circuits, as shown in Figure 3. A virtual circuit carries one call from its source to its destination on one physical connection, so the cells arrive in order at the destination. A connection is set up when the call arrives, and a virtual circuit is established through the network. A virtual path is established by the network to simplify routing by grouping virtual circuits [8].

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.

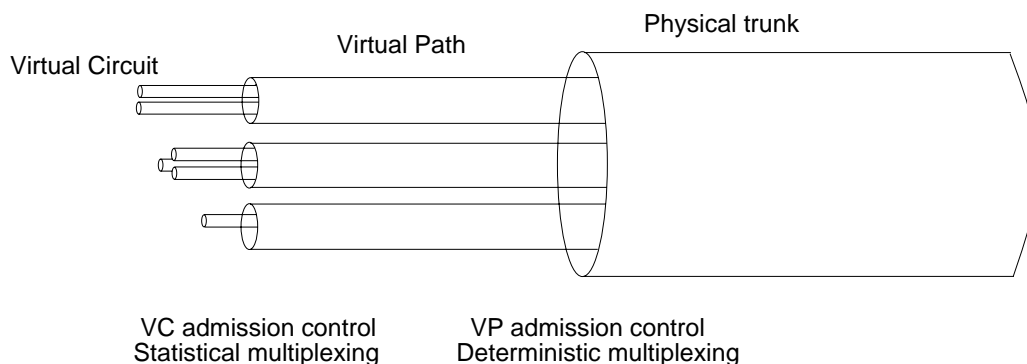


Figure 3: Virtual Path/Virtual Circuit connections

The virtual paths are carried by the physical network trunks, whose capacities are assumed to be fixed. A virtual path may be switched by a virtual path switch at intermediate nodes, where its virtual path identifier (VPI) may be changed. We assume that statistical multiplexing of the sources is done in the sense of allocating a capacity, C_{V_q} , to virtual path V_q which is less than the sum of the virtual circuit peak bandwidths using V_q . The network decides the capacities of the virtual paths and these satisfy the physical trunk capacity constraints. Here we assume deterministic multiplexing at the virtual path level, as in Figure 3.

To define our terminology, consider an example four node network with 150 Mb/s trunks as shown in Figure 4.

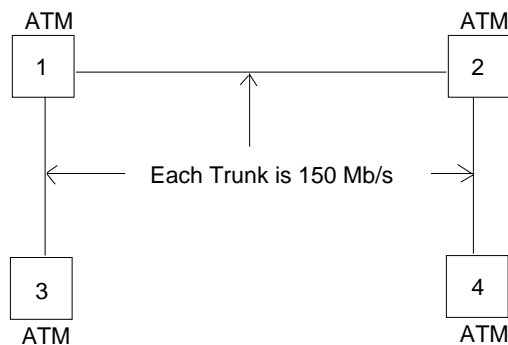


Figure 4: ATM Physical Network

This physical network contains an embedded logical network which consists of virtual paths connecting the ATM nodes. The ATM nodes are then virtual path switches for transit traffic and virtual circuit switches for originating and terminating traffic. A possible logical network is shown in Figure 5 for the physical network of Figure 4, where it can be seen that the physical trunk capacity constraints are satisfied.

The users connected to an ATM switch wish to send traffic to various destination ATM nodes. Their benefit versus bandwidth curve for a particular call is assumed to be concave increasing in general, as shown in Figure 6. 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 [6],[9].

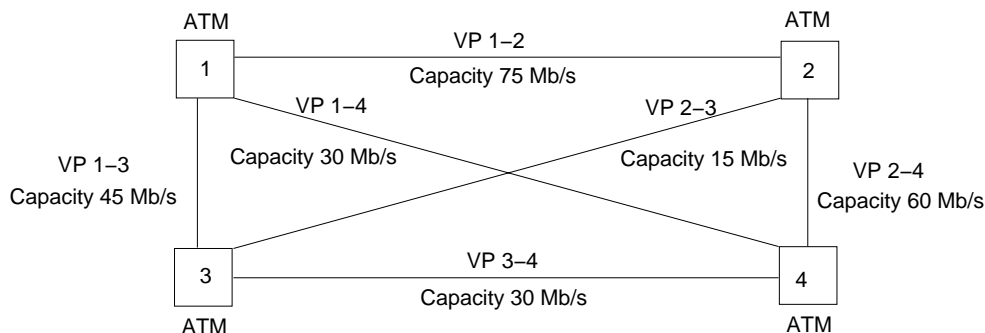


Figure 5: Logical Network embedded into Physical Network

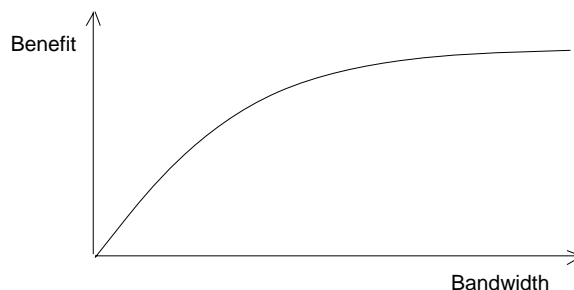


Figure 6: User Benefit as a function of Traffic Rate

Time is divided into pricing intervals. The users are free to alter their benefit functions from one pricing interval to the next, but these curves are fixed during a pricing interval. The ATM switch gives a price at the start of an interval which is also fixed for the duration of the interval. The user benefit function and the price then determine how many cells the user supplies during that interval. We are not concerned with the actual arrival distribution, as this is a cell scale component and can be buffered [10]. We concentrate on the burst scale component [10], and as long as the number of cells supplied in an interval is less than the total number that can be served in that interval we assume that there is no congestion.

3 PROBLEM FORMULATION

We assume that the physical network is given. The problem we consider is how to induce the users to contend with each other for network capacities so that the various constraints are satisfied. We divide this problem into two parts : firstly, how to get the users to share

a virtual path without exceeding its capacity; and secondly, how to assign the capacities of the virtual paths so that the physical trunk capacities are satisfied.

Consider the problem faced by a user attached to ATM node i in deciding their bandwidth demand for a particular call, denoted r , to destination ATM node j . Let the call be assigned to virtual path V_q . The network quotes all users a price per unit of bandwidth, π_q , on virtual path V_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 :

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

Each 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 [11]

$$\frac{\partial ben_{rq}}{\partial b_{rq}} - \pi_q = 0, \quad \forall r, V_q \quad (1)$$

Our previous assumptions on user behavior mean that, given the price per unit of bandwidth π_q , each user solves equation 1 to determine their optimal bandwidth demand for call r on virtual path V_q .

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 in fact no such system manager is needed. The system manager has to choose two sets of variables : user bandwidth demands $\{b_{rq}\}$ and virtual path capacities $\{C_{V_q}\}$. There are two types of constraints on the choice of values for these variables : virtual path capacities and physical trunk capacities.

The sum of virtual path capacities for all the virtual paths using a particular physical trunk, l_n , must be less than the physical trunk capacity :

$$\sum_{\text{all } V_q} C_{V_q} P_{qn} \leq C_n,$$

where P_{qn} is 1 or 0 depending on whether the virtual path V_q contains the physical trunk l_n or not. This capacity constraint can be enforced by using a barrier function approach, in which the constraint is dropped but a cost term $trunkcost_n$ corresponding to it is added to the system objective function. For example,

$$trunkcost_n := \frac{1}{C_n - \sum_{\text{all } V_q} C_{V_q} P_{qn}} - \frac{1}{C_n}$$

The shape of this cost term is shown in Figure 7, and ensures that as the physical trunk capacity is approached, the ‘cost’ of assigning virtual path capacities on the physical trunk becomes infinite.

The virtual path capacity constraint requires that the total cell arrival rate at a virtual path must be less than the capacity of that virtual path :

$$\sum_r b_{rq} \leq C_{V_q}, \quad \forall V_q$$

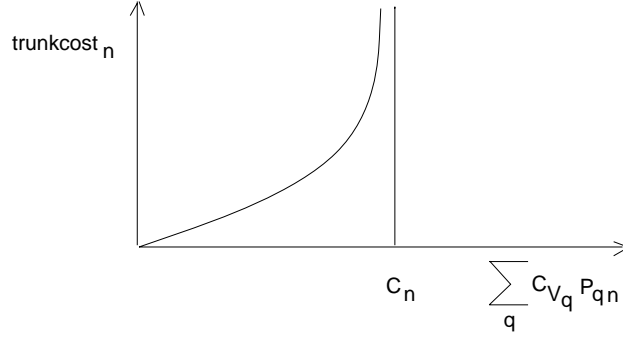


Figure 7: Barrier Function For Physical Trunk Capacity Constraint

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

$$Vcost_q := \frac{1}{C_{V_q} - \sum_r b_{rq}} - \frac{1}{C_{V_q}}$$

Note that the ‘cost’ term $Vcost_q$ depends on both the bandwidth demands from the users and the virtual path capacities. For a given virtual path capacity, increasing user bandwidth demands will increase $Vcost_q$; while for given bandwidth demands, increasing the virtual path capacity will decrease $Vcost_q$.

The system problem can be formulated as follows :

$$maximize \left\{ \sum_{all V_q} \left[\sum_r ben_{rq}(b_{rq}) \right] - Vcost_q \right\} - \sum_{all l_n} trunkcost_n$$

We assume that this problem has a solution. For strictly concave increasing benefit functions and strictly convex increasing cost functions, this solution is unique.

The optimality conditions for the system problem are [11]

$$\frac{\partial ben_{rq}}{\partial b_{rq}} - \frac{\partial Vcost_q}{\partial (\sum_r b_{rq})} \cdot 1 = 0, \quad \forall r, V_q \quad (2)$$

and

$$-\frac{\partial Vcost_q}{\partial C_{V_q}} - \sum_{all l_n} \frac{\partial trunkcost_n}{\partial (\sum_{all V_z} C_{V_z} P_{zn})} \cdot P_{qn} = 0, \quad \forall V_q \quad (3)$$

Comparison of equations 1 and 2 shows that by setting

$$\pi_q = \frac{\partial Vcost_q}{\partial (\sum_r b_{rq})} \quad (4)$$

individual user benefit maximization coincides with system optimality. 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 4 without any knowledge of the user benefit functions.

The virtual path capacities are chosen by the network operator to satisfy equation 3. This is equivalent to choosing the $\{C_{V_q}\}$ so that, for every virtual path V_q ,

$$-\frac{\partial V_{cost_q}}{\partial C_{V_q}} = \sum_{all\ l_n} \frac{\partial trunkcost_n}{\partial(\sum_{all\ V_z} C_{V_z} P_{zn})} \cdot P_{qn} \quad (5)$$

The term on the left-hand side of this equation is positive : the virtual path cost function is a decreasing function of the virtual path capacity. This term represents the marginal benefit to the network of assigning virtual path capacity C_{V_q} . The term on the right-hand side is also positive : as a virtual path capacity is increased, the physical trunk costs for those trunks which make up the virtual path also increase since there is less spare capacity on those trunks. This term represents the marginal cost to the network of assigning C_{V_q} .

4 DISTRIBUTED ALGORITHMS

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. T may be on the order of milliseconds using present technology, but its actual value is not crucial to the algorithms we propose. A lower bound on T is given by the minimum time required to observe the change in the user's demand. The price per unit of bandwidth π_q on virtual path V_q is announced by the network at the start of each pricing interval, and remains fixed for the duration of the interval. Hence, as explained in Section 3, 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.

Suppose the capacities of all the virtual paths are fixed. If the network operator announces prices $\{\pi_q\}$, the users optimally self-select and demand bandwidths $\{b_{rq}\}$. At each ATM node, these bandwidths are monitored, enabling the network operator to calculate the marginal cost of each virtual path with respect to its total traffic arrival rate, as given by the right-hand side of equation 4.

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. For example, if the user bandwidths are low enough that the virtual path marginal costs are lower than the prices, the network operator would lower the prices to try and increase the user bandwidths. If the virtual path marginal costs are higher than the prices, the user traffics are too high and the network operator would attempt to drive down the traffic rates by increasing the prices. Among the possible schemes for updating the prices are steepest descent and Newton's method. The updating scheme used will determine the convergence characteristics of the pricing algorithm.

We summarize this negotiation phase 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 virtual paths with respect to the bandwidths, as on the right-hand side of equation 4

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

Suppose now that the virtual path capacities are also variable. For given bandwidth measurements, the network operator can calculate the marginal cost of each virtual path with respect to its capacity, as given by the left-hand side of equation 5. For each virtual path V_q , the network operator can then check whether equation 5 is satisfied. If so, the system optimality conditions in equation 3 are satisfied for these values of the $\{b_{rq}\}$ and $\{C_{V_q}\}$. If not, then the network operator must adjust the values of the $\{C_{V_q}\}$ to try and satisfy equation 5. Again, there are a number of possibilities for the updating scheme that could be used.

The adjustments to the values of the $\{C_{V_q}\}$ cause changes in the user bandwidths. Hence these adjustments should be coordinated with the user negotiations. It is likely that the virtual path capacities would be updated much less frequently than the user bandwidths[8]. This leads to the two-phase algorithm outlined below.

Step 0. network operator chooses initial values of the virtual path capacities $\{C_{V_q}\}$

Step 1. network operator calculates marginal costs of the physical trunks with respect to these values of the $\{C_{V_q}\}$, as on the right-hand side of equation 5

Step 2. run the above Bandwidth Allocation Algorithm, UNTIL it is time to adjust the virtual path capacities \Rightarrow go to Step 3

Step 3. network operator calculates the marginal costs of the virtual paths with respect to the $\{C_{V_q}\}$, as on the left-hand side of equation 5

Step 4. network operator adjusts the virtual path capacities $\{C_{V_q}\}$ to reduce the difference between the terms in equation 5 \Rightarrow go to Step 1

If the user benefit functions remain constant, then if the above algorithm converges, it converges to the unique solution of the system problem. However in practice the user benefit functions may change every pricing interval, as the users alter their valuation of bandwidth depending on their service requirements and the quality of service provided by the network.

We envisage that this pricing algorithm runs continually in the network. Consequently the users are assumed to be responsive to new prices every T seconds. In practice we expect that the user responses would be automated, perhaps by programs running in their network interfaces. The users input their desired service characteristics to this program, as in the examples in Section 1. These characteristics may include monitoring the observed quality of service provided by the network. If this is not acceptable to the user, then effectively they increase their benefit function to take higher prices in the hope of getting a higher bandwidth. If the service quality is excellent, they may lower their benefit function in the hope of paying a lower price while still maintaining an acceptable level of service quality.

For example, suppose that a user wishes to send a file that consists of 100 cells and their required maximum delay is one second. If the pricing interval has length $T = 10msec$

then the user may start with a very low benefit curve. As this is very low the user only sends a few cells when the prices for the first interval are announced by the network. Then in the next interval they increase their benefit curve to see if they can get a better throughput. This can continue until towards the end of a one second period when the user starts to place a high benefit on getting their cells through and so will be willing to pay a higher price to get them accepted than initially. An intelligent network interface can be programmed with this kind of strategy.

The adjustment of virtual path capacities, which we assume takes place on a longer time scale than the user negotiations, is a network problem which requires the ATM nodes to communicate so that the physical trunk capacity constraints can be satisfied. This process may also be distributed. Each ATM node could be made responsible for assigning virtual path capacities $\{C_{V_q}\}$ for those virtual paths originating at the node. The embedding of virtual paths into the physical network would then determine how the ATM nodes coordinate to ensure that the physical trunk capacity constraints are met.

5 VPN TRAFFIC MANAGEMENT

An emerging application for ATM networks is the setting up of Virtual Private Networks, or VPNs. In this case one operator controls the network and all applications running on it. These applications may include voice, video conferencing, video telephony and data transfer. In some VPNs, the operator may place a high priority on operational data transfer, and low priority on other data transfers, leaving the delivery guarantee up to the higher layer protocols such as TCP. Also the relative importance of voice and video applications may vary over time and possibly from node to node. For example, operational 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 get higher loss even though it may not be able to tolerate cell loss as much 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 be assigned depending on the present state of the network, as well as previous network states. Rather than have a complicated priority scheme, a pricing scheme like the one proposed in this paper could be used. The benefit functions would be set by the operator for the different services such as voice or video. The operator could also set different benefit functions for applications of the same type. Each such application would then input traffic according to its assigned benefit function and the current state of the network, as reflected in the prices.

Another use of pricing is in dropping cells due to congestion. Suppose that data transfer 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 as the whole frame has to be retransmitted anyway. 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.

6 CONCLUSIONS AND FUTURE DIRECTIONS

This paper suggests pricing and user self-regulation as a means of allocating bandwidth in ATM networks. The suitability of this scheme for traffic management, and its compatibility with other bandwidth allocation and congestion control methods proposed in the literature, is still an open question. If user benefit functions do not have the shape assumed in this paper, the convergence of the pricing algorithm in quasi-static conditions may be affected. Our assumption 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. Fixed network costs may be recovered using an access charge rather than the usage-based charges in our pricing scheme. However we believe that the possibility of using dynamic pricing for bandwidth allocation is an interesting one and worthy of further investigation.

Acknowledgements

We would like to thank Pravin Varaiya, Gustavo de Veciana and Mike Wong at UC Berkeley, and Padmanabhan Srinagesh and Mike Honig at Bellcore, for helpful discussions about this work. The first author acknowledges Charles McCorkell of Dublin City University for his continuing support.

References

- [1] M. de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Ellis Horwood, 1991.
- [2] V. E. Benes, "Programming and control problems arising from optimal routing in telephone networks", *Bell Syst. Tech. J.*, vol. 45, pp. 1373-1438, 1966.
- [3] 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, 1993.
- [4] S. M. Walters, "A new direction for broadband ISDN", *IEEE Commun. Mag.*, pp. 39-42, Sept. 1991.
- [5] F. P. Kelly, "Tariffs, policing and admission control for multiservice networks", *10th UK Teletraffic Symp.*, 1993.
- [6] A. de Fontenay, M.H. Shugard and D. S. Sibley, *Telecommunications Demand Modeling: an integrated view*, North-Holland, 1990.
- [7] 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.
- [8] J. Burgin and D. Dorman, "Broadband ISDN resource management: The role of virtual paths", *IEEE Commun. Mag.*, pp. 44-48, Sept. 1991.
- [9] US Department of Commerce, *Telecommunications in the Age of Information*, Oct. 1991.
- [10] J. W. Roberts, "Variable-bit-rate traffic control in B-ISDN", *IEEE Commun. Mag.*, pp. 50-56, Sept. 1991.
- [11] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd Ed., Addison-Wesley, 1984.