Telecommunication Systems 18:4, 331–358, 2001

Optimal Policies for ATM Cell Scheduling and Rejection *

EROL GELENBE
School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816-2450, USA

VIJAY SRINIVASAN
CoSine Communications, 1200 Bridge Pkwy, Redwood City, CA 94065, USA

SRIDHAR SESHADRI sseshadr@stern.nyu.edu
Department of Statistics and Operations Research, Leonard N. Stern School of Business, New York University, NY 10012-1118, USA

NATARAJAN GAUTAM ngautam@psu.edu
Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, PA 16802, USA

Received January 1998; Revised December 2000

Abstract. This paper addresses the following questions related to buffer management schemes for high speed integrated services networks: (i) given the pattern of cell arrivals from different classes of traffic, can buffer control significantly influence the effect of cell loss, and (ii) what are the "best" policies for selecting cells for transmission from buffers in the network nodes as well as for rejecting cells when the buffers are full. The basic approach to answering these questions is to impute a cost of losing cells which could depend on the class of application, and to minimize this cost over the finite or infinite time horizons. The policies we derive using this cost minimization approach are best in the sense that they minimize linear cost functions of cell losses, at each instant of time during the system's operation. We also show how to construct policies that minimize general cost functions of cell loss rates.

Keywords: QoS, loss minimization, scheduling

1. Introduction

In this paper we consider buffer management policies for very high speed asynchronous transfer mode (ATM) switches that support multiple classes of traffic and guarantee negotiated Quality of Service (QoS) to the users [8]. The explosive growth of Internet, users and applications with many carriers supporting ATM networks [7], large ATM switches with capacity over 40 Gb/s or 100 Gb/s are becoming a necessity [31]. Given

* This work has been partially supported by Duke University via a J.B. Duke Fellowship, and by the IBM Corporation.

the very high speeds of emerging ATM switches, the difficulty in accurately characterizing network traffic, and, the necessity to provide stringent QoS, it is extremely important to develop buffer management policies that:

- uses little or no information about the state of the buffers (at the ports) in the switch as well as the state of neighboring switches,
- works in a (near) optimal fashion for any traffic arrival process (i.e., policy independent of traffic generation), and,
- can be implemented with little or no change to the existing switching and scheduling mechanisms.

The scheduling policies developed in this paper addresses all the above concerns.

Many ATM switch manufacturers such as Marconi (Fore) [38] have implemented a large number of suggestions made in the literature including increasing the size of the output buffers (current Marconi switches have capacity of 10 Gb/s with output buffers of 65,536 cells per port) to avoid excessive loss of packets. There are several reasons why packet loss is still a concern. Firstly, extremely bursty traffic which is often seen in IP can cause severe losses even with such large buffers. Under bursty conditions, one way of performing congestion control is to use an adaptive-cell discard policy [8]. It is shown (in [25]) that by discarding only 5% of the traffic, the queue lengths can be reduced by a factor of 10. The scheduling policies we develop in this paper are adaptive, i.e., packet loss is monitored and packets are admitted or rejected on this basis. Second, several authors have suggested that congestion control could be achieved through pricing the network traffic based on service requirements (see [27,49]). Such pricing schemes only control admission but do not help scheduling (see [27]). In this paper we develop cost-based scheduling policies so that users receive the service they paid for.

Third, large buffers can reduce loss but will increase delay. The recent literature focuses on employing a shared buffer for all the ports of the switch as this minimizes buffer size requirement, improves buffer utilization and optimizes cell loss performance [31]. A hierarchical buffer management scheme for an ATM switch with shared buffer is developed in [21] where a combination of static and dynamic thresholds are used to ensure efficient and fair usage of buffer memory. A survey of the buffer management methods that have been proposed for shared memory ATM switches is presented in [1]. Sharma and Viniotis [46] consider a shared buffer systems and derive the properties of the optimal policy which gives the least-average expected total cell loss probability under a given discrete Batch Markovian Arrival Process scheme.

Although shared buffer ATM switches have desirable properties, it is too large and complex to be easily implemented [31]. Alternatively making each buffer smaller and shared among a group of ports results in small-sized buffers that are easy to implement. The Knockout switch developed by AT&T was an early prototype where the amount of buffers was reduced at the cost of higher cell loss [20].

The standard queue scheduling structures available in most switches include: static priority, weighted round robin, and, waited fair queueing, all of which will not change in small or large capacity switches [31]. These queue scheduling structures are described in detail (including mathematical models) in [30]. In this paper we focus on small sized buffers and suggest optimal scheduling policies that are easily implementable extensions to the standard queue scheduling structures available on most switches. There are several other papers in the research literature that deal with buffer management in ATM switches [9,32,37,56].

We model a buffer in an ATM multiplexer (switch) as a finite queue with a single server. Cells sharing this buffer arrive to the multiplexer from several different traffic streams with different levels of tolerance to cell loss. A buffer management algorithm exercises control over the queue through two main actions: the *scheduling rule* determines the order of transmission of cells from the buffer, and the *rejection rule* is used to decide which cell(s) to discard from the buffer in the event of congestion. In this paper, the goal of the buffer management policies is to optimize a given loss-oriented performance criteria. When several classes of service are present, a multi-class performance criterion can be designed by constructing a suitable cost function depending on the desired effect. The simplest performance criterion that we consider is the weighted sum of the number of lost cells of each stream, over a given time interval. Presumably, this weighted sum will reflect the relative importance of cells of different traffic classes, and the weights assigned to each class will be chosen in accordance with the priority given to that class. More complex performance criteria that address issues of fairness and contracted loss guarantees are also considered.

For a single node in an ATM network, the main contributions of this paper are:

- We consider an arbitrary arrival sequence of cells from multiple streams to a single switch with a finite buffer. Each stream i, carries a different per unit cost, c_i, associated with cell loss. The obvious control policy which always schedules the highest cost cell, and rejects the least cost cell minimizes the cost function, $\sum_i c_i L_i(t)$, where $L_i(t)$ is the number of cells from stream i lost over the interval $[0, t]$. This result is proved to hold over every finite time interval for arbitrary sequence of cell arrivals.
- Again we consider an arbitrary arrival sequence of cells from multiple streams to a single switch with a finite buffer. The cost in this case is expressed as a general function of the cell loss rates, $\lim_{t\to\infty} L_i(t)/t$. For this model, we develop a numerical optimization procedure for constructing control policies that provide a performance as close to the optimal as desired.
- We completely characterize the set of achievable loss vectors over a finite time interval, and the set of achievable loss rates over the infinite time horizon.

The novel features of these results are:

- A cost-based approach to finding good buffer management algorithms is shown to provide flexible means for trading-off cell losses amongst different service classes.
- All of the results are formulated for arbitrary sequences of cell arrivals and therefore do not rely upon any stochastic models. This is very attractive for ATM networks in particular, since we make no special assumptions about traffic.

The study of buffer management mechanisms for a single node is useful in providing better insights into the much harder end-to-end (or network-wide) case. In order to illustrate the single node results, and to demonstrate its applicability in the end-to-end case, we present simulations of an ATM network carried out using a commercial-off-the-shelf networking software COMNET [14] on a personal computer. The ATM network is modeled as a sequence of switches supporting different traffic streams, both through (end-to-end) and cross traffic. The simulation results indicate that the optimal single-node-policies work well when used for end-to-end control in a network.

There is a considerable amount of literature related to the work presented in this paper. The problem of minimizing cell loss in the single node case has been addressed in [3,12,13,27–29,34,36,40–43,47,48]. In [28], two classes of traffic are modeled and the cell loss is analyzed by modeling the switch as an $M/G/1$ queueing system. The scheduling rule is First Come First Served (FCFS) and the rejection rule allows a cell from the high priority class to displace a cell from the low priority class when the buffer is full. Kroner et al. [34] analyze the same system but consider another scheme called partial buffer sharing. In this scheme, cells from a class are blocked from entry when the buffer occupancy reaches a certain threshold level. They also propose route separation in which buffer space is reserved for different classes along a route in the network. Cidon et al. [12] are concerned with buffering policies for two classes of traffic that can guarantee a loss probability to cells from the high priority class regardless of the arrival characteristics of cells from the low priority class. They assume that the cells are of the same size and need a deterministic time for transmission. They fix the scheduling rule to be FCFS for this two class system and seek a rejection rule that can guarantee a level of performance for the high priority class which is at least as good as when this class is served by a dedicated switch with a given buffer of size M. Lin and Silvester [36] also model a two class system, and evaluate the performance under different priority queueing policies. They model the system as a discrete-time multichannel queue with finite buffer capacity. The cell transmission time is deterministic; and arrival as well as departure instants correspond to discrete time points called slot boundaries. Their model permits correlation between the arrival processes of the two classes. Bounds are provided for the loss probability of the high and low priority classes. They show that cell loss can be significantly influenced by using priority schemes for buffer control. Petr and Frost [42] use a dynamic programming approach to minimize the average cost of discarding cells in a system with an arbitrary number of classes of traffic – given that there is waiting room for just one cell. The service time is assumed to be constant and the arrival pattern is as in [36]. This study is extended in [43] where the problem of maximizing the offered load over all nested threshold-type policies is considered.

In [48], a single switch with constant transmission times and an arbitrary arrival process is considered. It is shown that when there are two classes of traffic, a rule that gives priority to the high priority cells minimizes the loss of high priority cells. These results are a special case of our results. Awater and Schoute [3] consider both delay and losses. They consider a slotted-time model with constant transmission times, and assume that the number of cells that arrive in a time slot is given by an i.i.d. multinomial distribution. The novel feature of their paper is that one traffic stream requires low-delay and the other requires low-loss. They characterize the optimal policy using Markov decision theory.

Peha and Tobagi [40,41] use a cost function-based approach to minimize cell loss as well as delay. They propose cost-based scheduling and cost-based dropping heuristics to fine tune network performance, and evaluate the performance of these heuristics using simulation. A key contribution of their work is to show that complex buffer control schemes can actually be implemented in hardware in a cost-effective manner. Clare and Rubin [13] consider a slotted-time queueing system in which the service time is deterministic and cells can begin and end transmission only at slot boundaries. For each source, the number of cells that arrive in a slot form an i.i.d. sequence, and, the arrivals of cells from different sources in the same slot may be correlated. They show that a work conserving policy, i.e., one that never idles when there are cells awaiting transmission and a non-expulsive rejection policy, i.e., one that rejects a cell only when the buffer is full, minimizes the total number of lost cells at every time instant (see remark 3 in section 3 for an application of this result).

The remainder of this paper is organized as follows. Section 2 describes the model we consider, the definitions, and notation used. The analytical result for linear cost functions is presented in section 3 which is followed by simulations to support the analytical results. Optimization of general cost functions is discussed in section 4. In this section, we also describe how the results can be used to construct practical controls. Numerical examples based on simulations using COMNET to extend the analytical results from the single node to the network case are given in section 5.

2. The model and definitions

We study a *single* multiplexer or switch within a route R, which has the a topology as shown in figure 1. The route R traverses K switches $S_0, \ldots, S_{K-1}$. The *main* traffic stream originates at the first switch and traverses each of the K switches in R. The main stream can be some mixture of N traffic streams. The architecture of a switch is shown in figure 2. The switch fabric is assumed to be fully interconnected. Demultiplexers at the input feed into multiplexers that implement output queueing. Each of the switches has J input links and J output links. The traffic arriving on each of the J input links is demultiplexed and queued at output buffers at the J output links. The route R uses one input link and one output link in each switch. The output buffers have a fixed capacity of B cells. A cell arriving at the switch on any of the other $J-1$ input links uses the same output link from the switch as route R with probability $1/J$. These cells constitute the *cross* traffic that interferes with the cells from the main stream. Cells from the cross traffic arrive to the same output buffer used by main stream cells, and leave the switch on the same output link as route R.

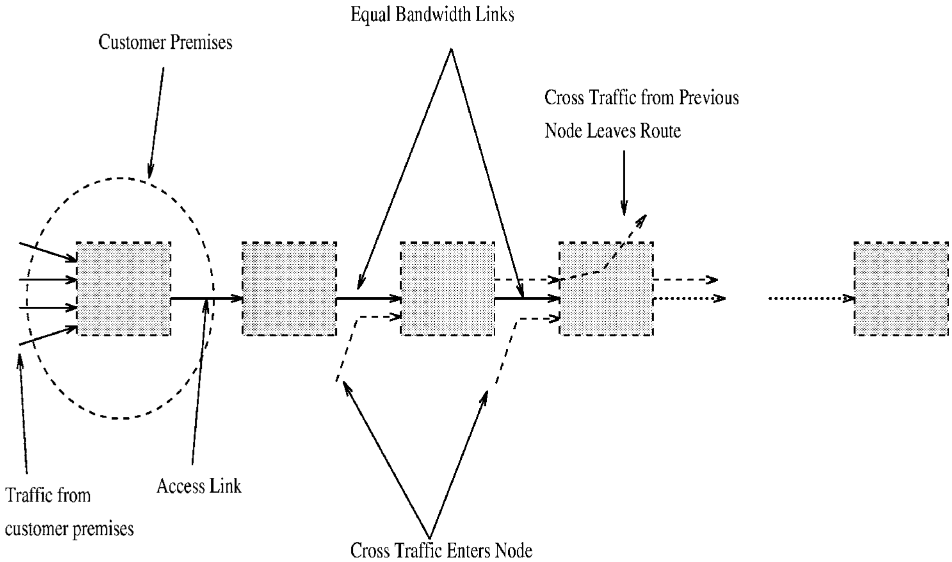

Figure 1. The route R under study.

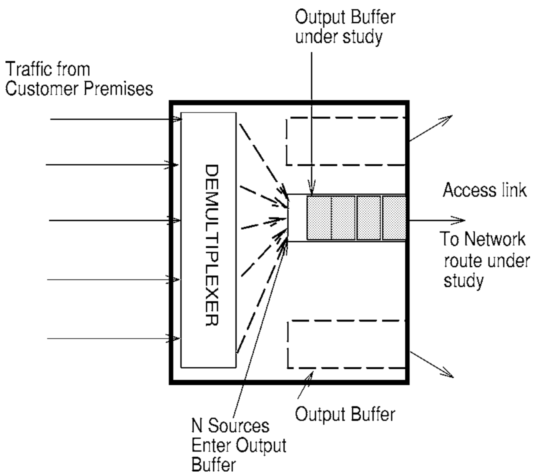

Figure 2. Switch architecture.

The multiplexer is modeled as a single server queue with a finite buffer having waiting room for B cells. Cells from M different streams arrive to the multiplexer. We next describe the parameters of this system:

$M =$ number of different arrival streams of cells.

$T_{n,i} =$ inter-arrival time between the nth and the $(n+1)$st cell from stream i to the system, $n = 0, 1, 2, \ldots, i = 1, 2, \ldots, M$. The $T_{n,i}$'s do not depend on the state of the system. This is the only restriction placed on the arrival processes.

$S_n =$ service time of the nth cell to be taken up for transmission. We assume that the service times S_n's do not depend on the inter-arrival times.

$F_t =$ the history of the arrivals, past service times and residual service time at time t, if any.

$\mathcal{A} =$ class of scheduling rules considered. The scheduling rules we consider are non-idling (work conserving, [54]), non-anticipatory (cannot know the future), non-preemptive (will not stop ongoing work), and non-expulsive (will not reject a cell unless the buffer is full).

We make the qualification of being non-anticipatory precise below (please see the definition of $u(t)$). We further define:

$$\begin{aligned} L_m &= \text{the time instant at which the } m\text{th cell is lost,}\\ L(t) &= \text{total number of cells (from all streams) lost till time } t,\\ u(t) &= \text{the control exercised on the system.} \end{aligned}$$

The control involves (a) deciding which cell to take up for transmission when the server is free and there are cells available for transmission in the buffer and (b) deciding which cell to drop when an arriving cell finds the buffer full. The controls considered are those that use only the information contained in the history till time t, i.e., F_t. Such controls are said to be non-anticipatory. For convenience we denote the use of the control $\{u(s),\ 0 \leqslant s \leqslant t\}$ by the superscript u:

$$\begin{aligned} L_i^u(t) &= \text{total number of cells of type } i \text{ lost till time } t\text{, when control } u \text{ is used, } i = 1, 2, \ldots, M,\\ N_i^u(t) &= \text{the number of type } i \text{ cells in the system at time } t,\ i = 1, 2, \ldots, M,\\ N(t) &= \sum_{i=1}^{M} N_i^u(t)\ \forall u \in \mathcal{A}. \end{aligned}$$

In our model, the scheduling rules are non-expulsive and work conserving, the inter-arrival times do not depend on the state of the system, and the service times do not depend on the identity of the arrival stream. Therefore it follows that $N(t)$, L_m, and $L(t)$ do not depend on the control u, see [13] for a proof. It can also be verified that given a control $u(t)$, the quantities $L_i^u(t)$ and $N_i^u(t)$ can be obtained from using just the information contained in the history till time t, i.e., F_t.

3. Linear cost

In this section we address the problem of determining a control u in order to minimize the weighted sum of the $L_i^u(t)$'s, i.e., minimize over $u \in \mathcal{A}$, $\sum_{i=1}^{m} c_i L_i^u(t)$, for all t. Before considering this problem, we prove an important property of cell loss under a small class of controls, called the absolute priority rules. It turns out that for optimization under the performance criteria studied in this paper, it is sufficient to restrict our attention to this class. We present the analytical results first, followed by simulations to support the analytical results.

3.1. Analytical results

Consider the class of scheduling and rejection rules, Π, in which a rule $\pi \in \Pi$, first, orders the stream indices as $(\pi_1, \pi_2, \ldots, \pi_M)$, then assigns the highest scheduling priority to stream π_1, next highest to stream π_2 and so on with the lowest scheduling priority being given to stream π_M. The rule also assigns the lowest rejection priority to the stream π_1, and the highest to stream π_M. When choosing to schedule the next cell or when forced to reject a cell, the rule π always chooses a cell from the stream with the highest scheduling or rejection priority respectively. The class Π is said to be the class of absolute priority rules, and $\Pi \subset \mathcal{A}$. We show below that *any* rule from the class Π achieves a particular ordering (see lemma 1) of the number of cells lost till time t, for every t. This property can then be used to optimize the cost of lost cells for suitable loss functions, by simply selecting an appropriate rule from Π.

Lemma 1. Starting at time 0, from the same initial conditions, for any $\pi \in \Pi$,

$$\sum_{i=1}^{n} \left(L^{\pi}_{\pi_i}(t) - L^{u}_{\pi_i}(t)\right) \leqslant 0, \quad \forall n = 1, 2, \ldots, M,\ \forall t \geqslant 0,\ \text{and any } u \in \mathcal{A}.$$

Proof. Define

$$X_n(t) = \sum_{i=1}^{n} \left(N^{\pi}_{\pi_i}(t) - N^{u}_{\pi_i}(t)\right)$$

and

$$Y_n(t) = \sum_{i=1}^{n} \left(L^{u}_{\pi_i}(t) - L^{\pi}_{\pi_i}(t)\right), \quad n = 1, 2, \ldots, M.$$

We show that the following relationship (which is stronger than the claim of the lemma) is true for all time t and all $n = 1, 2, \ldots, M$:

$$Y_n(t) \geqslant \max\left(X_n(t), 0\right). \tag{1}$$

Fix n. The relation in (1) is true at time 0, since the initial conditions are the same. Assume that the relation is true in the interval $[0, L_m]$, i.e., from time 0 to the instant of the mth cell loss. Due to the nature of the scheduling rule (i.e., π schedules a cell from stream π_i ahead of π_j if $i < j$), the relation is seen to hold during the time $[L_m, L_{m+1})$. At time L_{m+1}, we need to consider two distinct cases:

Case 1: Rule π rejects a cell of stream $j \notin \{1, 2, \dots, n\}$ or both u and π reject a cell from stream $j \in \{1, 2, \dots, n\}$. In this case, inequality (1) holds at time L_{m+1} as it held at time L_{m+1}^-.

Case 2: Rule π rejects a cell from stream $j \in \{1, 2, \dots, n\}$ and u rejects a cell of stream $j \notin \{1, 2, \dots, n\}$. It follows that:

$$Y_n(L_{m+1}) = Y_n\bigl(L_{m+1}^-\bigr) - 1 = Y_n(L_m) - 1. \tag{2}$$

Since inequality (1) held at time L_m, we also have

$$Y_n(L_m) - 1 \geqslant X_n(L_m) - 1 \tag{3}$$

and

$$X_n(L_{m+1}) = X_n\bigl(L_{m+1}^-\bigr) - 1 \leqslant X_n(L_m) - 1. \tag{4}$$

The equality in (4) is obtained from the assumption for case 2, and the inequality by the nature of the rule π. Further note that if $X_n(L_m) \leqslant 0$, then case 2 can never happen. To see this, assume that $X_n(L_m) \leqslant 0$. This implies that at time L_m, the system controlled using π had fewer cells belonging to the streams $\{1, 2, \dots, n\}$ than the system controlled using u. As no cells were rejected during $[L_m, L_{m+1})$, and due to the nature of the scheduling rules, the system controlled using π must still have a smaller total number of cells belonging to the streams $\{1, 2, \dots, n\}$ at time L_{m+1}^-. This implies that once again by the nature of the two scheduling rules and because the total number of cells belonging to all streams that are present in the buffer at any time t, does not depend on the scheduling rule, π could not have rejected a cell from the streams $\{1, \dots, n\}$ at time L_{m+1}. Thus we must have:

$$X_n(L_m) > 0. \tag{5}$$

Combining (2)–(5), we get:

$$Y_n(L_{m+1}) = Y_n(L_m) - 1 \geqslant X_n(L_m) - 1 \geqslant \max\bigl(X_n(L_{m+1}), 0\bigr). \tag{6}$$

□

Using this lemma, we can now prove:

Theorem 2. Given constants $c_1 \geqslant c_2 \geqslant c_3 \geqslant \cdots \geqslant c_M$, if the objective is to minimize over $u \in \mathcal{A}$, $\sum_{i=1}^{m} c_i L_i^u(t)$, for all t, then the objective can be achieved by using the rule $\pi \in \Pi$, such that $\pi_1 = 1, \pi_2 = 2, \dots, \pi_M = M$.

Proof. We use an inductive proof. Assume that given any $b_1 > b_2 > \cdots > b_n > 0$, $\sum_{i=1}^{n} b_i L_i^\pi(t) \leqslant \sum_{i=1}^{n} b_i L_i^u(t)$ is true. This is trivially true when $n = 1$. Then given $b_1 > b_2 > \cdots > b_n > b_{n+1} > 0$, consider

$$\sum_{i=1}^{n+1} b_i\bigl(L_i^\pi(t) - L_i^u(t)\bigr) = \sum_{i=1}^{n}(b_i - b_{n+1})\bigl(L_i^\pi(t) - L_i^u(t)\bigr) + b_{n+1}\sum_{i=1}^{n+1}\bigl(L_i^\pi(t) - L_i^u(t)\bigr). \tag{7}$$

The first term on the right hand side of (7) is less than or equal to 0 by the induction hypothesis. The second term is also nonpositive by lemma 1. □

Remarks.

1. The control in theorem 2 achieves the minimum cost at all times t. Such a control has been called by other researchers to be a *least control* or a *path-wise least cost control* (see [50,55], for examples).
2. The control exhibited in theorem 2 will minimize all functions, $f(\cdot)$, of cell loss that have the property:

$$\sum_{i=1}^{n}(x_i - y_i) \leqslant 0, \quad \forall n = 1, 2, \ldots, M-1, \quad \text{and}$$
$$\sum_{i=1}^{M}(x_i - y_i) = 0 \quad \text{implies} \quad f(x_1, x_2, \ldots, x_M) \leqslant f(y_1, y_2, \ldots, y_M).$$

3. In the ATM context, the service time can be assumed to be the deterministic. Given this, it can be shown that if the loss function is nondecreasing in the number of cells lost from each stream, and if the objective is to minimize the loss function for a *single* switch, then it is never optimal to idle the server when there is work to be done or expel a cell when the buffer is not full, see [13]. Thus the qualifications that the control policy should be work conserving and non-expelling can be dropped in the ATM context.
4. A buffer management algorithm that always selects higher priority cells for transmission before the lower priority ones, and discards low priority cells when the queue is full, can be implemented efficiently in hardware. An architecture that applies the concepts of fully distributed and highly parallel processing to arrange the cells' transmission or rejection order is proposed in [10].
5. Lemma 1 completely characterizes the set of achievable loss vectors over a finite time interval.

3.2. Simulation examples

In this section, we present some simulation results to support the analytical results presented in the previous section as well as extend the results from a single node case to a network. We first discuss the experimental set up and then describe the results.

3.2.1. Experimental set-up

In these examples, the performances under the optimal policy and the First In First Out (FIFO) policy are compared. We consider a "main stream" of traffic which travels over route R, consisting of a series of switches with finite buffers. "Cross traffic" with respect to this specific main-stream traffic arrives to the same output buffer used by cells from the main traffic stream, and departs from the switch on the same output link as route R; but upon arrival to the next switch, the cross traffic is demultiplexed to a different output buffer from route R and no longer interferes with the main stream (see figure 1). Time is slotted with the length of a slot equal to the time to transmit a cell. A cell transmitted by switch S_{i-1} takes one slot time to arrive at switch S_i. A cell arriving at a switch is immediately queued at the output buffer for transmission to the next switch. However, this transmission can only occur at the earliest in the next time slot.

3.2.2. Numerical results

We model the main traffic stream and the cross traffic streams as Poisson arrival processes. Other traffic stream models are considered for the general cost structure case in section 4. The buffer size at all switches is $B = 40$. There are $K = 8$ switches on the route. We compare FIFO policy and the optimal policy. There are two classes of cells. The linear cost function used is $C(t) = 2L_1(t) + L_2(t)$, where $L_i(t)$ is the total number of cells of type i lost till time t. The optimal policy is to use an absolute priority rule which gives class 1 the priority for scheduling and rejection over cells from class 2. The FIFO rule gives neither class priority over the other, and schedules and rejects cells based on the time of arrival to the buffer. Neither rule distinguishes between cells from the main traffic stream and the cross traffic stream. The fraction of class 1 (high priority) traffic and class 2 (low priority) traffic is varied from 10 to 90%. The traffic load is maintained at a high enough level to correspond to a loss probability of the order of 10^{-3}. This level may be much higher than in an actual network, but it has the advantage of testing the policies under extreme conditions with short simulation runs. The main traffic traffic intensity is 0.75 and the cross traffic intensities are all 0.22. Each simulation run is of length 10^6 slots, and we performed 30 replications of the simulations. The length of the simulations and the number of replications were chosen such that the 95% confidence interval for cell loss approximately equaled 5–10% of the average cell loss.

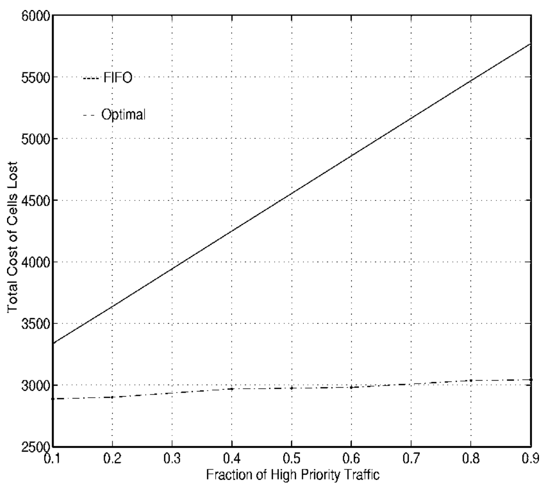

Figure 3. Total cost of cells lost under linear cost structure.

Figure 3 shows the total cost of cell loss as a function of the mix of high and low priority traffic. We observe that the cost curve under the optimal policy is almost flat, indicating that the loss vector is relatively insensitive to changes in the traffic mix, and most of the cell loss is borne by class 2 traffic. The total cost of using the optimal is well below the FIFO cost.

4. General cost

Although the cost functions discussed in the previous sections are intuitively appealing, there are instances where other cost functions would be of interest. As an example, for a two class system *class 1* could be far more critical with respect to cell loss because it is an emergency communication channel, and *class 2* may tolerate cell loss, $L_2(t)$, as long as it does not exceed some value $\lambda t > 0$. This would lead to a cost function of the form:

$$a_1\big[L_1(t)\big]^2 + a_2\big[L_2(t) - \lambda t\big]^+, \tag{8}$$

where a_1, a_2 are positive constants and $[x]^+ = x$ if $x > 0$, and $[x]^+ = 0$ if $x \leqslant 0$. Other examples in which an appropriate cost functions could be used in the context of ATM communication are enumerated below.

- Minimize the maximum loss of cells from any stream to ensure fairness,[1]

$$\min\big\{\max_i\{L_i(t)\}\big\}.$$

- Ensure that the loss for each stream i does not exceed a contracted fraction, say α_i, of the total number of lost of cells,

$$\min\sum_i\bigg[L_i(t) - \alpha_i\bigg[\sum_j L_j(t)\bigg]\bigg]^+.$$

- Use a combination of the above criteria. For example, provide two grades of service such that each grade as a whole does not exceed a given fraction of the total number of lost cells, and within each grade minimize the maximum loss of cells over its constituent streams.

4.1. Analytical results

It is impossible in general to obtain a control that minimizes the cost for each time t, when the cost functions are general. Therefore, we restrict our attention to the characterization of optimal control policies when the performance measure can be expressed as a

[1] This corresponds to the max–min criteria for measuring fairness. See, for example, the ATM Forum Technical Committee's Traffic Management Specification 4.0 (April, 1996) which outlines fairness criteria with regard to bandwidth (section I.3, p. 73). The max–min criteria figures prominently amongst the examples discussed in these specifications.

function of the cell loss rates.[2] Assume that there exist non-negative finite real numbers β_i^u's such that

$$\lim_{t\to\infty} \frac{L_i^u(t)}{t} = \beta_i^u, \quad i = 1, 2, \dots, M. \tag{9}$$

The numbers β_i^u's denote the average rate of cell loss for type i cells when control u is used. The set of achievable loss rates are characterized below. Let, $\beta_i^u(t) = L_i^u(t)/t$.

Lemma 3. The vector of loss rates, $\{\beta_1, \beta_2, \dots, \beta_M\}$, satisfies the conservation law,

$$\sum_{i=1}^{M} \beta_{\pi(i)}^{\pi}(t) = \sum_{i=1}^{M} \beta_{\pi(i)}^{u}(t), \tag{10}$$

and

$$\sum_{i=1}^{n} \beta_{\pi(i)}^{\pi}(t) \leqslant \sum_{i=1}^{n} \beta_{\pi(i)}^{u}(t), \quad \forall n \leqslant M. \tag{11}$$

Proof. The proof follows from lemma 1 and equation (9). □

Based on this lemma, the search for the optimal performance over the infinite time horizon can be reduced to solving a deterministic optimization problem.

Theorem 4. If the objective is

$$\min_u f\big(\beta_1^u, \beta_2^u, \dots, \beta_M^u\big), \quad u \in \mathcal{A},$$

the optimal value of the problem can be found by solving the mathematical program:

$$\min_{x\in P_\infty} f(x_1, x_2, \dots, x_M) \tag{12}$$

where, $x \in P_\infty$ is a performance vector which can be expressed as a convex combination of the loss rates under the absolute priority rules.

Proof. From [45, theorem 1] and earlier work in [26] every achievable performance vector is contained in the convex polytope P_∞ (which is the base of a polymatroid) generated by the performance vectors $\{\beta_1^\pi, \beta_2^\pi, \dots, \beta_M^\pi\}$ corresponding to the absolute priority rules ($\pi \in \Pi$). □

Remarks.

1. The practical significance of theorem 4 is two fold: (i) the minimum over the polytope P_∞ gives an achievable value of the target performance measure, and (ii) it

[2] The ATM Forum Technical Committee's Traffic Management Specification 4.0, specifies the Cell Loss Ratio (CLR) as one of the QoS parameters that form the traffic contract. This supports our modeling the cost as a function of the loss rate of cells.

suggests that the optimal performance is a weighted average of the performance under the different absolute priority rules. Given this solution, the optimal performance can be obtained either exactly or as closely as desired (see section 4.2) by determining a suitable combination of absolute priority rules to use. This two step numerical optimization procedure makes the implementation of a good control policy easier than when the control has to remember the entire history of arrivals and service times.

2. To highlight the difference in the nature of a good control policy for a network vis-a-vis that for a single node, consider the same route R as shown in figure 1. Assume that there is *no* cross-traffic, i.e., all external traffic comes to the first node in the route. Assume that the service times are deterministic and equal at all switches. Under these assumptions, all losses take place at the first node in the system. Therefore the optimal policy for a single node will also be optimal for the entire system. However, when there is cross traffic, the situation can change substantially because feedback from adjacent nodes regarding the level of congestion at those nodes may help us determine a better control. In particular the optimal control may no longer be work conserving. This issue will be addressed in future work.
3. If there are large transmission delays between switches in a network, it would be virtually impossible to use feedback information from up stream switches in order to improve the control at a single switch. This is typically true since the switching times have become extremely small in giga-bit type ultrahigh-speed switches. In such circumstances, the control technique suggested in this paper will perform equally well in the network context.
4. Lemma 3 completely characterizes the set of achievable loss rates over the infinite time horizon.

4.2. Constructing an optimal control

To construct a control using theorem 4, we need to first determine the vector of cell loss (viz., the extreme points of the polytope P_∞) for each absolute priority rule. Simulation can be used for this purpose. However, since the loss probability is often of the order 10^{-9} in ATM environments, the simulation runs will have to be very long unless fast simulation techniques based on importance sampling are used [39]. If it were possible to characterize the arrival patterns before hand – e.g., from packetized voice – the calculations of cell loss can be performed off-line and the actual control determined in real time. Another method of estimating losses is to use approximations. However, with the available approximation techniques for queues with bounded buffer space (see [6; 26, chapter 4; 33; 44]), losses cannot be computed accurately. Simulations suggest that estimates of the relative losses of cells from different streams for a given absolute priority rule can be extremely sensitive to the arrival pattern of cells, also see the survey given in [2]. The problem of obtaining estimates for the loss probability under the absolute priority rules for general traffic input patterns is addressed in our previous work

[23,24,35] and others' [4,5,15,16,18,19,51] based on large deviations, effective bandwidths and their extensions.

After obtaining the polytope and solving the mathematical program, the control policy can be determined using one of two methods. (i) For a system with Poisson arrivals and i.i.d. service times, once the optimal performance vector **x** has been obtained from solving the program shown in (12), the control policy may be determined by using the methods of [22]. (ii) The problem of obtaining the optimal control for the general case is a very hard problem. Therefore, we suggest the use of *different absolute priority rules over different time slices* to obtain as close an approximation to the optimal solution as desired. In fact for the numerical example for a 2-class traffic system (section 5 the policy of using an absolute priority for one class for a time slice and then switching the priorities for the next time slice and so on is found to perform far better than other scheduling policies. To see the validity of such a procedure, notice that starting with n packets in the buffer the losses for any stream i under a given policy u will not exceed $L_i^u(t) + n$ when compared to the cell loss starting from an empty buffer. Therefore, as the optimal performance measure is a weighted combination of the performance under different absolute priority rules, the performance under this procedure can be made as close to the optimal solution as desired.

4.3. Simulation experiments

We will demonstrate using a numerical example (generated by simulation experiments) how to construct an optimal control. We need to validate the analytical results described so far. We also have to illustrate how the policy works when the single node results are extended to a network (see remark 2 of section 4.1). Therefore we simulate an ATM network using a commercial network simulation software COMNET [14]. In addition, we also use the simulations to test the performance of the scheduling policies.

COMNET can be used to simulate traffic for a route R depicted in figure 1. Multiple classes of mainstream and cross stream traffic can be simulated, and performance measures in terms of cell-loss and delay can be generated for different control/scheduling policies. To make the comparisons meaningful, the input traffic is kept the same for all control/scheduling policies by using the same seed values. The number of switches (K), route (R), number of input/output links to switches (J), output buffer capacity (B), and, number of traffic streams N, all defined in section 2 can be set in COMNET.

For the numerical examples in section 5, we use the route R described in section 3.2, $K = 8$ switches, $J = 2$ output links to switches, buffer capacity $B = 40$, $N = 2$ traffic streams for each of the main and cross streams. Also, the traffic is such that class 1 cells constitute 20% of the total traffic and class 2 traffic 80% for both the main as well as the cross streams. In addition, the cell size, link speeds and scheduling/control mechanism (especially FIFO/FCFS and piecewise constant absolute priority rules) can be chosen as appropriate.

To demonstrate that the analytical results in sections 3 and 4 do not depend on the traffic arrival processes, we consider several types of traffic models that are supported by COMNET. In the next two sections, we describe three traffic models and their effect on the performance measures.

4.3.1. Traffic models

We consider three models from a suite of traffic generation models available in COMNET (the numerical values described here are used throughout the remainder of this paper):

- *Poisson*. The cells generated by sources are modeled using a Poisson process where the time between two cells generated are according to an exponential distribution. Note that the Poisson traffic model is not considered appropriate due to the long-range dependence observed in the network traffic [53]. However, in order to demonstrate that the analytical results are not dependent on the traffic model as well as to benchmark against other models we use this Poisson model. The numerical values chosen for the inter-arrival times for class 1 mainstream, class 2 main stream, class 1 cross stream and class 2 cross streams are 6.67, 1.67, 5.681 and 22.73 time units, respectively.
- *Pareto*. The cells generated by sources are modeled using a renewal process where the time between two cells generated are according to a Pareto distribution which is heavy-tailed (infinite variance). Although not long-range dependent in the strictest sense, using a Pareto distribution for the inter arrival times between the cells represents the network traffic reasonably well. Moreover, in terms of benchmarking, this falls somewhere in between the Poisson and long-range dependent traffic performance. The numerical values chosen for the Pareto inter-arrival time distribution parameters (k, β) for class 1 mainstream, class 2 main stream, class 1 cross stream and class 2 cross streams are (2.5, 1.6), (0.625, 1.6), (2.1307, 1.6) and (8.523, 1.6), respectively, in appropriate time units.
- *Bursty*. The cells generated by sources are modeled using a bursty on-off process where a random number of cells are "burst" out when the source is on and a random number of cells are slowly generated when the source is off. The number of cells burst out or generated are according to a normal distribution. However, the on and off time durations are heavy-tailed (Pareto distribution). It is known that on–off sources with heavy-tailed on-time and off-time distributions can be used to capture essential features of self-similarity observed in measurements [17,52]. The version of COMNET used does not support any other model for long range dependent traffic. The numerical values chosen for the bursty traffic are described in table 1.

See figure 4. For each traffic model (Poisson, Pareto and bursty), for a single sample path, the number of cells generated during a 1000 unit of time interval is obtained from the simulations. Observations are made at times 1000, 2000, . . . , 10000 that record the number of cells generated during the previous 1000 time units. Note that all traffic models use the same mean cell generation rate (represented by the solid line). Note that the Poisson traffic is relatively smooth around the mean in comparison to the Pareto and bursty traffic. Clearly the Pareto and bursty traffic represent some of the Internet traces

Table 1
Numerical values for the bursty traffic model.

	On burst		On time		Off burst		Off time	
	μ	σ	k	β	μ	σ	k	β
Class 1 mainstream	30	8	1	1.5	6	1	9.375	1.6
Class 2 mainstream	7.5	2	0.25	1.5	1.5	0.25	2.34375	1.6
Class 1 cross stream	10	3	1.25	1.5	1	0.25	9.375	1.6
Class 2 cross stream	40	12	5	1.5	4	1	37.5	1.6

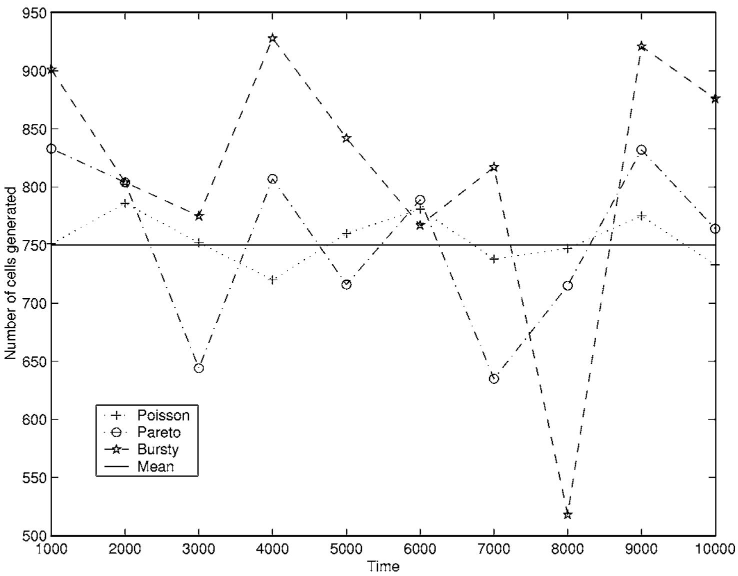

Figure 4. Comparing the number of cells generated every 1000 time units for the three traffic streams: Poisson, Pareto and bursty of a single sample path.

pretty accurately. Clearly the bursty traffic has a lot higher variability than the Pareto traffic.

4.3.2. Effect of traffic models

Before using the three traffic models to analyze the scheduling policies (this is done in section 5), we compare the performance of the three traffic models with each other in terms of the average loss. Note that when the average arrival rates are kept the same and the traffic intensity or load is kept the same, the total loss incurred due to the different burstiness in the traffic is very different. In fact the loss observed in the bursty model case is several times larger than the Pareto model case which in turn is several times larger than the Poisson case. Note that the performance measure that is compared is the total

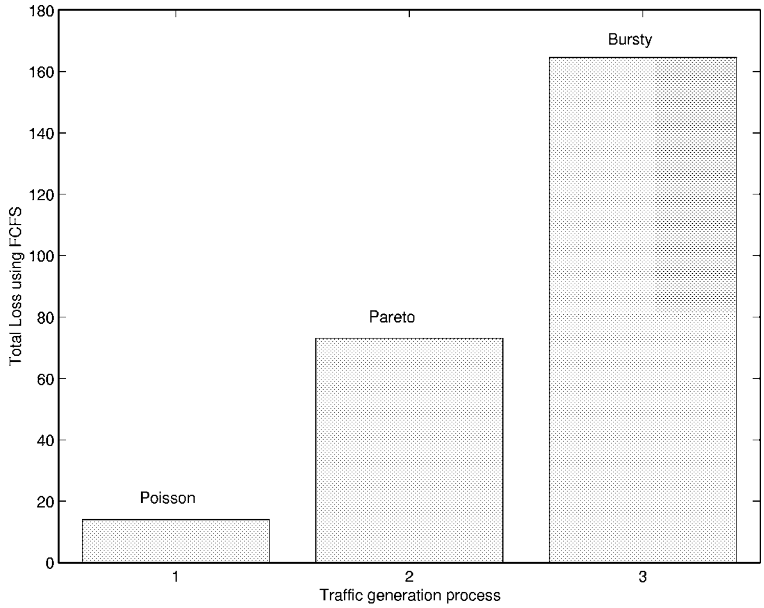

Figure 5. Comparing the total loss under FCFS for the three traffic streams: Poisson, Pareto and bursty under identical traffic and load conditions.

loss experienced by all classes of traffic under only the FCFS scheduling mechanism. However the losses for the individual classes as well as losses under other scheduling schemes exhibit similar behavior as observed in figure 5.

Although the total loss is significantly different for the different traffic models, the relative performance between different policies for scheduling multi-class traffic does not depend on the traffic models. This is illustrated in section 5. To make the analysis comparable and meaningful, simulation experiments whose results are reported in section 5, are performed by keeping the total loss rate approximately the same order of magnitude for all the three traffic models. This is done by suitably changing the link speeds. Increasing the link speeds reduces the losses considerably.

5. Results for convex cost structure

In this example, we illustrate the use of ideas from theorem 4. Let the cost function be $C(t) = (L_1(t)^2 + L_2(t)^2)$. With the use of the FCFS rule, the average cell losses for the two classes of traffic and the cost for the first 10000 slots are as shown in tables 2, 6 and 10 for Poisson, Pareto and bursty traffic, respectively. (All figures for cell loss reported in this section are averages over 25 replications.[3]) To apply theorem 4, we need to first generate the loss vectors for the two possible absolute priority rules. When class 1 is given absolute priority over class 2, we observe the results given in tables 3, 7

[3] The standard error was 5%.

Table 2
Cell losses with FCFS rule for Poisson traffic with load = 0.97.

Time	# of cells lost Class 1	# of cells lost Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	2.92	12.44	163.2800
2000	6.28	27.96	821.2000
3000	9.32	40.48	1725.4928
4000	13.16	55.80	3286.8256
5000	16.36	69.36	5078.4592
6000	20.72	82.52	7238.8688
7000	23.40	91.64	8945.4496
8000	26.96	103.72	11484.6800
9000	29.72	113.44	13751.9120
10000	32.92	127.00	17212.7264

Table 3
Cell losses with priority for class 1 under Poisson traffic with load = 0.97.

Time	# of cells lost Class 1	# of cells lost Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	0	15.68	245.8624
2000	0	34.92	1219.4064
3000	0	50.48	2548.2304
4000	0	69.68	4855.3024
5000	0	86.76	7527.2976
6000	0	104.60	10941.1600
7000	0	116.40	13548.9600
8000	0	132.44	17540.3536
9000	0	145.44	21152.7936
10000	0	162.28	26334.7984

Table 4
Cell losses with priority for class 2 under Poisson traffic with load = 0.97.

Time	# of cells lost Class 1	# of cells lost Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	17.20	0	295.8400
2000	35.52	0	1261.6704
3000	52.12	0	2716.4944
4000	71.56	0	5120.8336
5000	88.08	0	7758.0864
6000	106.24	0	11286.9376
7000	117.36	0	13773.3696
8000	133.48	0	17816.9104
9000	147.40	0	21726.7600
10000	163.16	0	26621.1856

Table 5
Cell losses with equal priority for Poisson traffic under load = 0.97.

Time	# of cells lost Class 1	# of cells lost Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	0	16.96	287.6416
2000	18.04	16.96	613.0832
3000	18.04	33.48	1446.3520
4000	36.44	33.48	2448.7840
5000	36.44	51.60	3990.4336
6000	53.68	51.64	5544.1024
7000	53.68	63.60	6926.5024
8000	69.52	63.60	8977.9904
9000	69.52	77.96	10910.7920
10000	85.32	77.96	13357.2640

Table 6
Cell losses with FCFS rule for Pareto traffic with load = 0.87.

Time	# of cells lost Class 1	# of cells lost Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	2.08	8.44	75.5600
2000	4.36	18.00	343.0096
3000	6.12	25.68	696.9168
4000	8.92	38.56	1566.4400
5000	11.84	49.04	2545.1072
6000	13.48	57.24	3458.1280
7000	15.48	67.68	4820.2128
8000	17.24	74.56	5856.4112
9000	19.80	83.04	7287.6816
10000	21.40	91.60	8848.5200

Table 7
Cell losses with priority for class 1 under Pareto traffic with load = 0.87.

Time	# of cells lost Class 1	# of cells lost Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	0	10.72	114.9184
2000	0	22.80	519.8400
3000	0	32.48	1054.9504
4000	0	48.48	2350.3104
5000	0	61.88	3829.1344
6000	0	72.04	5189.7616
7000	0	84.56	7150.3936
8000	0	93.56	8753.4736
9000	0	104.68	10957.9024
10000	0	115.04	13234.2016

Table 8
Cell losses with priority for class 2 under Pareto traffic with load = 0.87.

Time	# of cells lost Class 1	Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	11.56	0	133.6336
2000	23.80	0	566.4400
3000	33.44	0	1118.2336
4000	49.84	0	2484.0256
5000	62.80	0	3943.8400
6000	73.40	0	5387.5600
7000	86.04	0	7402.8816
8000	95.92	0	9200.6464
9000	107.04	0	11457.5616
10000	116.68	0	13614.2224

Table 9
Cell losses with equal priority for Pareto traffic under load = 0.87.

Time	# of cells lost Class 1	Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	0	11.44	130.8736
2000	11.60	11.44	265.4336
3000	11.60	21.72	606.3184
4000	27.12	21.72	1207.2528
5000	27.12	35.20	1974.5344
6000	37.40	35.20	2637.8000
7000	37.40	47.92	3695.0864
8000	46.44	47.92	4453.0000
9000	46.44	59.84	5737.4992
10000	56.08	59.84	6725.7920

Table 10
Cell losses with FCFS rule for bursty traffic with load = 0.77.

Time	# of cells lost Class 1	Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	2.56	10.72	121.4720
2000	6.28	24.12	621.2128
3000	9.60	35.08	1322.7664
4000	13.12	45.00	2197.1344
5000	16.64	56.96	3521.3312
6000	19.68	66.64	4828.1920
7000	22.28	77.52	6505.7488
8000	25.04	86.92	8182.0880
9000	29.48	102.40	11354.8304
10000	33.68	113.64	14048.3920

Table 11
Cell losses with priority for class 1 under bursty traffic with load = 0.77.

Time	# of cells lost Class 1	Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	0	13.56	183.8736
2000	0	30.80	948.6400
3000	0	45.32	2053.9024
4000	0	58.96	3476.2816
5000	0	74.52	5553.2304
6000	0	87.60	7673.7600
7000	0	101.36	10273.8496
8000	0	113.64	12914.0496
9000	0	133.80	17902.4400
10000	0	149.44	22332.3136

Table 12
Cell losses with priority for class 2 under bursty traffic with load = 0.77.

Time	# of cells lost Class 1	Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	14.60	0	213.1600
2000	31.36	0	983.4496
3000	46.20	0	2134.4400
4000	60.52	0	3662.6704
5000	76.00	0	5776.0000
6000	90.12	0	8121.6144
7000	103.96	0	10807.6816
8000	116.52	0	13576.9104
9000	136.20	0	18550.4400
10000	151.48	0	22946.1904

Table 13
Cell losses with equal priority for bursty traffic under load = 0.77.

Time	# of cells lost Class 1	Class 2	Cost $L_1(t)^2 + L_2(t)^2$
1000	0	14.12	199.3744
2000	16.80	14.12	481.6144
3000	16.80	28.92	1118.6064
4000	30.68	28.92	1777.6288
5000	30.68	44.48	2919.7328
6000	44.04	44.48	3917.9920
7000	44.04	58.80	5396.9616
8000	55.40	58.80	6526.6000
9000	55.40	80.00	9469.1600
10000	70.68	80.00	11395.6624

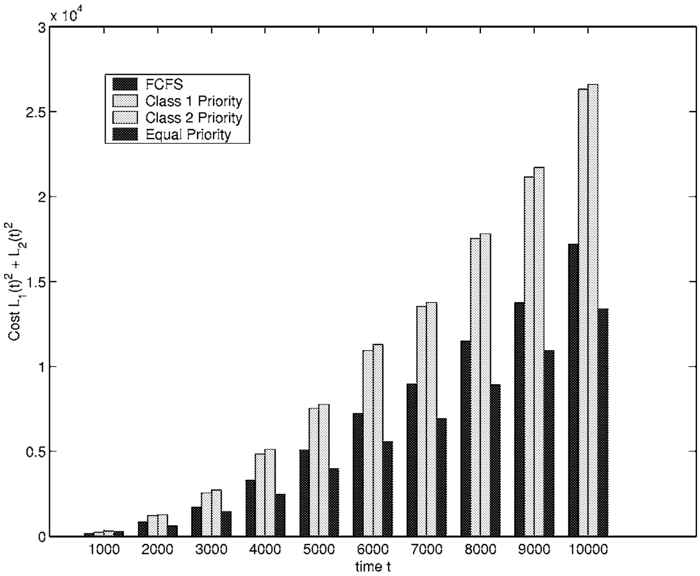

Figure 6. Comparing the convex cost function for the Poisson arrival pattern under four scheduling policies: FCFS, priority to class 1, priority to class 2, and equal priority.

and 11 for Poisson, Pareto and bursty traffic, respectively. The cell losses when class 2 is given priority over class 1, are shown in tables 4, 8 and 12 for Poisson, Pareto and bursty traffic, respectively.

Using these results we can construct the polytope of theorem 4. For this example, the polytope is the straight line joining two points in the (X, Y)-plane, that are located approximately at $(a, 0)$ and $(0, a)$, where a is the total number of cells lost till that point in time. For example, at time 1000 in tables 11 and 12 for the bursty traffic, the line connects the points $(0, 14)$ and $(14, 0)$ approximately. Applying theorem 4, the optimal solution is to give equal priority to classes 1 and 2. The rule is implemented by alternately giving priority to the two classes in time slices of 1000 time units. Applying this rule, yields the data given in tables 5, 9 and 13 for Poisson, Pareto and bursty traffic, respectively. This example apart from illustrating the use of theorem 4 also serves to underscore the non-intuitive fact that even though the traffic from class 1 is much smaller than that from class 2, for a symmetric convex loss function, the optimal solution is to give "equal" priority to both classes.

Figures 6–8 compare the convex cost function for Poisson, Pareto and bursty traffic models respectively under the four scheduling schemes: FCFS, priority to class 1, priority to class 2 and equal priority. Clearly the performance of equal priority is the best followed by the FCFS scheme. Both perform much better than the priority rules. In tables 2–13 the breakdown of the respective losses and cost functions for the different scheduling policies are illustrated for Poisson (tables 2–5), Pareto (tables 6–9) and bursty (tables 10–13) traffic models. Note that for figures 6–8 as well as tables 2–13, the traffic model parameters are as reported in section 4.3. However, in order to keep the

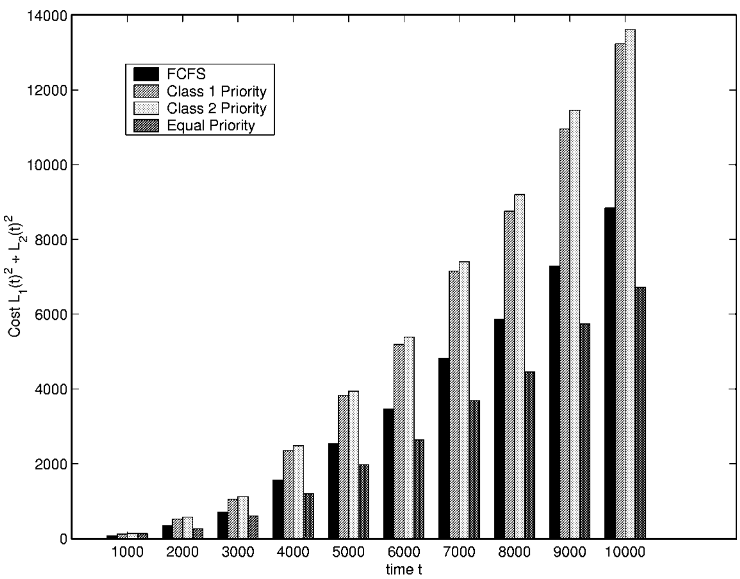

Figure 7. Comparing the convex cost function for the Pareto arrival pattern under four scheduling policies: FCFS, priority to class 1, priority to class 2, and equal priority.

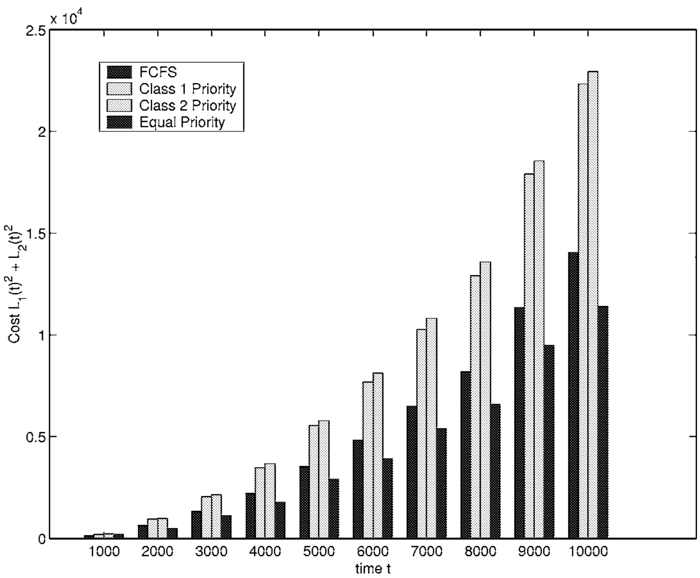

Figure 8. Comparing the convex cost function for the bursty arrival pattern under four scheduling policies: FCFS, priority to class 1, priority to class 2, and equal priority.

losses to about the same order of magnitude, the link speeds are modified so that the load using the Poisson, Pareto and bursty traffic models are 0.97, 0.87 and 0.77, respectively.

Although the analytical models only suggested that the equal priority policy will be optimal over a single node, we find via simulation experiments that even the end-to-end performance of the equal priority policy is clearly superior than the other scheduling policies – FIFO, priority to class 1 and priority to class 2. However, the simulation run times of all the policies (FIFO, priority to class 1, priority to class 2 and the optimal equal priority) were exactly the same.

Implementation

In terms of implementation, the equal priority policy can be implemented on ATM switches using the same logic as that for COMNET by assigning equal weights to the two classes of traffic in a weighted round-robin scheme (the scheme is explained in [30] and is available on most ATM switches). Since the weighted round-robin scheme is work conserving, when the scheduler is on class 1 (and draining out class 2 traffic using any leftover capacity) it is equivalent to giving class 1 high priority and class 2 lower priority. When the scheduler toggles to class 2, then class 2 receives higher priority. Note that other policies that result out of a different objective function, can be implemented by appropriately modifying the different service scheduling policies available in ATM switches.

6. Conclusions

In this paper, we have completely characterized the set of achievable performance vectors with regard to cell loss over the finite time horizon, and with respect to cell loss rate over the infinite horizon, for a single switch with finite buffer. We have also developed a procedure for determining the optimal performance under any performance criteria which can be expressed either as a linear function of the cell loss or an arbitrary function of the cell loss rates, and indicated a method for determining controls that can yield a performance as close as desired to the optimal value. The analytical results are validated and extended from a single switch to a network of switches using a commercial-off-the-shelf network simulation software package. We consider a 2-class traffic example with objective function as the sum of the squares of the cumulative cell loss over time. For this example we show by running simulation experiments with varying traffic models that the optimal policy which assigns equal priorities to the two classes by alternating the priorities, performs far better on an end-to-end basis than other policies such as FIFO and absolute priority. In future work, we intend to study the queueing and loss processes when the switch is operated using the controls suggested in this paper, with the aim of developing approximations for the loss probabilities. Other performance measures such as latency and delay-jitter will be investigated in the future. Various Quality of Service requiring traffic will also be analyzed. This analysis will be extended to IP-based networks in future work.

Acknowledgements

The authors thank the associate editor and the reviewer for suggestions that led to considerable improvements in the content and presentation of this paper.

References

[1] M. Arpaci and J.A. Copeland, Buffer management for shared-memory ATM switches, `http://www.comsoc.org/pubs/surveys/1q00issue/copeland.html` (2000).
[2] S. Asmussen and R.Y. Rubinstein, Steady state rare events simulation in queueing models and its complexity properties, in: *Advances in Queueing: Theory, Methods, and Open Problems*, ed. J.H. Dshalalow (CRC Press, Boca Raton, FL, 1995).
[3] G.A. Awater and F.C. Schoute, Optimal queueing strategies for fast packet switching of mixed traffic, IEEE Journal on Selected Areas in Communication 9(3) (April 1991) 458–467.
[4] A.W. Berger and W. Whitt, Effective bandwidths with priorities, IEEE/ACM Transactions on Networking 6(4) (August 1998).
[5] A.W. Berger and W. Whitt, Extending the effective bandwidth concept to network with priority classes, IEEE Communications Magazine (August 1998).
[6] J.A. Buzacott and J.G. Shanthikumar, *Stochastic Models of Manufacturing Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1993).
[7] C. Catalano, Asynchronous transfer mode, Computerworld (3 January 2000).
[8] S. Catanzariti, New ATM service targets IP networks, Network World (1 June 1998).
[9] C. Chan, S. Hu, P. Wang and Y. Chen, A FIFO-based buffer management approach for the ATM GFR services, IEEE Communications Letters 4(6) (2000).
[10] H.J. Chao, A novel architecture for queue management in ATM networks, in: *Proc. of GLOBECOM*, Phoenix, AZ, 1991, pp. 1611–1615.
[11] K.L. Chung, *A Course in Probability Theory* (Academic Press, New York, 1974).
[12] I. Cidon, R. Guerin and and A. Khamisy, On protective buffer policies, IEEE/ACM Transactions on Networking 2(3) (June 1994) 240–246.
[13] L.P. Clare and I. Rubin, Performance boundaries for prioritized multiplexing systems, IEEE Transactions on Information Theory 33(3) (May 1987) 329–340.
[14] COMNET III – Simulating application and network performance, `http://www.compuware.com/products/ecosystems/comnet/`.
[15] G. de Veciana, C. Courcoubetis and J. Walrand, Decoupling bandwidths for networks: A decomposition approach to resource management, in: *INFOCOM'94*, pp. 466–473.
[16] G. de Veciana, G. Kesidis and J. Walrand, Resource management in wide-area ATM networks using effective bandwidths, IEEE Journal on Selected Areas in Communications 13(6) (1995) 1081–1090.
[17] N.G. Duffield and W. Whitt, Network design and control using on-off and multi-level source traffic models with heavy-tailed distributions, in: *Self-Similar Network Traffic and Performance Evaluation*, eds. K. Park and W. Willinger (2000).
[18] A.I. Elwalid, D. Heyman, T.V. Lakshman, D. Mitra and A. Weiss, Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing, IEEE Journal on Selected Areas in Communications 13(6) (1995) 1004–1016.
[19] A.I. Elwalid and D. Mitra, Analysis, approximations and admission control of a multi-service multiplexing system with priorities, in: *INFOCOM'95*, 1995, pp. 463–472.
[20] S. Fahmy, A survey of ATM switching techniques, `http://www.cs.purdue.edu/homes/fahmy/` (2000).
[21] R. Fan, A. Ishii, B. Mark, G. Ramamurthy and Q. Ren, An optimal buffer management scheme with dynamic thresholds, in: *GLOBECOM*, Vol. 1b, 1999, pp. 631–637.

[22] A. Federgruen and H. Groenvelt, Characterization and optimization of achievable performance in queueing systems, Operations Research 36(3) (1988) 733–741.
[23] N. Gautam and V. G. Kulkarni, Applications of SMP bounds to multi-class traffic in high-speed networks, to appear in Queueing Systems (December 2000).
[24] N. Gautam, V.G. Kulkarni, Z. Palmowski and T. Rolski, Bounds for fluid models driven by semi-Markov inputs, Probability in the Engineering and Informational Sciences 13 (1999) 429–475.
[25] N. Gautam and S. Seshadri, Approximations for system performance under self-similar traffic, in: *Proc. of the 3rd Internat. Conf. on Telecommunications and E-commerce* (November 2000) pp. 239–250.
[26] E. Gelenbe and I. Mitrani, *Analysis and Synthesis of Computer Systems* (Academic Press, London, 1980).
[27] L. Georgiadis, R. Guérin and A. Parekh, Optimal multiplexing on single link: Delay and buffer requirements, in: *Proc. of IEEE INFOCOM*, Toronto, Canada, June 1994.
[28] G. Hebuterne and A. Gravey, A space priority queueing mechanism for multiplexing channels, Computer Networks and ISDN Systems 20(1–5) (December 1990) 37–43.
[29] Y.-H. Jeon, Conservation laws and multiplexing schemes for delay and loss requirements in broadband networks, Ph.D. thesis, Department of Electronic and Computer Engineering, North Carolina State University, Raleigh, NC (1992).
[30] G. Kesidis, *ATM Network Performance*, 2nd ed. (Kluwer Academic, Dordrecht, 2000).
[31] K.Y. Kou, Realization of large-capacity ATM switches, IEEE Communications Magazine 37(12) (December 1999).
[32] B. Kraimeche, A buffer management and channel allocation strategy for QoS support in an ATM switch, in: *9th Internat. Conf. on Computer Communications and Networks* (2000) pp. 463–466.
[33] V. Kriman and R.Y. Rubinstein, Polynomial time algorithms for estimation of rare events in queueing models, in: *Frontiers in Queueing: Models and Applications in Science and Engineering*, ed. J.H. Dshalalow (CRC Press, Boca Raton, FL, 1997).
[34] H. Kroner, G. Hebuterne, P. Boyer and A. Gravey, Priority management in ATM switching nodes, IEEE Journal on Selected Areas in Communications 9(3) (April 1991) 418–427.
[35] V.G. Kulkarni, and N. Gautam, Admission control of multi-class traffic with service priorities in high-speed networks, Queueing Systems 27 (1997) 79–97.
[36] A.Y.-M. Lin and J.A. Silvester, Priority queueing strategies and buffer allocation protocols for traffic control at an ATM integrated broadband switching system, IEEE Journal on Selected Areas in Communications 9(9) (December 1991) 1524–1536.
[37] A. Mokhtar and M. Azizoglu, A random early discard framework for congestion control in atm networks, in: *Proc. of IEEE ATM Workshop* (1999) pp. 45–50.
[38] Marconi, `http://www.marconi.com` (2000).
[39] S. Parekh and J. Walrand, A quick simulation method for excessive backlogs in networks of queues, IEEE Transactions on Automatic Control 34(1) (1989) 54–66.
[40] J.M. Peha and F.A. Tobagi, A cost-based scheduling algorithm to support integrated services, in: *Proc. of IEEE INFOCOM*, Bal Harbour, FL, April 1991, pp. 741–753.
[41] J.M. Peha and F.A. Tobagi, Implementation strategies for scheduling algorithms in integrated-services packet-switched networks, in: *Proc. of GLOBECOM*, Phoenix, AZ, December 1991, pp. 1733–1740.
[42] D.W. Petr and V.S. Frost, Optimal packet discarding: An ATM-oriented analysis model and initial results. in: *Proc. of IEEE INFOCOM*, San Francisco, CA, June 1990, pp. 537–542.
[43] D.W. Petr and V.S. Frost, Nested threshold cell discarding for ATM overload control: Optimization under cell loss constraints, in: *Proc. of IEEE INFOCOM*, Bal Harbour, FL, April 1991, pp. 1403–1412 (12A.4).
[44] K.W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks* (Springer, London, 1995).
[45] J.G. Shanthikumar and D.D. Yao, Multiclass queueing systems: Polymatroidal structure and optimal scheduling control, Operations Research 40 (May/June 1992).

[46] S. Sharma and Y. Viniotis, Optimal buffer management policies for shared-buffer ATM switches, IEEE/ACM Transactions on Networking 7(4) (August 1999).

[47] Y. Takagi, S. Hino and T. Takahashi, Priority assignment control of ATM line buffers with multiple QOS classes, IEEE Journal on Selected Areas in Communications 9(7) (September 1991) 1078–1092.

[48] L. Tassiulas, Y. Hung and S.S. Panwar, Optimal buffer control during congestion in an ATM network node, IEEE/ACM Transactions on Networking 2(4) (August 1994).

[49] D. Walker, F.P. Kelly and J. Solomon, Tariffing in the new IP/ATM environment, Telecommunications Policy 21 (1997) 283–295.

[50] L. Wein, Optimal control of a two-station Brownian network, Mathematics of Operations Research 15(2) (1990) 215–242.

[51] W. Whitt, Tail probabilities with statistical multiplexing and effective bandwidths for multiclass queues, Telecommunications Systems 2 (1993) 71–107.

[52] W. Whitt, Performance models consistent with traffic measurements, `http://www.research.att.com/w̃ow/performance.html` (2000).

[53] W. Willinger and V. Paxson, Where mathematics meets the Internet, Notices of the American Mathematical Society 45(8) (September 1998) 961–970.

[54] R. Wolff, *Stochastic Modeling and the Theory of Queues* (Prentice-Hall, Englewood Cliffs, NJ, 1989).

[55] P. Yang, Least controls for a class of constrained linear stochastic systems, Mathematics of Operations Research 18(2) (May 1993) 275–291.

[56] X. Yijun and J. Kanginakudru, CLR in multicast ATM switches with shared memory and buffer management, in: *GLOBECOM*, Vol. 2, 1999, pp. 1391–1395.