A fair buffer allocation scheme

Juha Heinanen and Kalevi Kilkki
Telecom Finland
P.O. Box 228, SF-33101 Tampere, Finland
E-mail: juha.heinanen@tele.fi

Abstract

An appropriate service for data traffic in ATM networks requires large buffers in network nodes. However, large buffers without a proper allocation scheme may lead to an unsatisfactory quality of service. Most present allocation schemes either necessitate a complicated queueing system or they do not offer a sufficient fairness. This paper describes a rather simple buffer management scheme that results in fair allocation of bandwidth among competing connections by using only a FIFO buffer. The performance and fairness of the allocation scheme has been analysed by means of a simulation program.

1. Introduction

Asynchronous Transfer Mode (ATM) is the basis for future high-speed telecommunication networks. The strength of ATM lies in its superior flexibility which enables a wide variety of services and applications to be efficiently integrated in one network. One of the main difficulties of ATM is related to management of the network. Especially the control of multiple types of traffic with different service requirements has proved to be very difficult.

There are different strategies for controlling an ATM network. The simplest one is to use the peak-rate allocation method with an appropriate Usage/Network Parameter Control (UPC/NPC) scheme (see e.g., [1]). However, this approach is somewhat inconsistent with the original idea of ATM, which implies the possibility to exploit the statistical behaviour of traffic streams. Especially in case of data traffic, peak-rate allocation may lead to very low utilisation.

Another approach, called *rate envelope multiplexing* [2], the operator endeavours to ensure that the combined input rate of multiplexed connections is within the envelope, at least with a very high probability. Rate envelope multiplexing is usually based on small FIFO buffers which allows a relatively simple implementation of network nodes. This approach has several drawbacks: the statistical properties of connections could be very difficult to predict, the achievable utilisation remains low if the peak rate of sources increases beyond a small fraction of the link rate, and during an overload situation, the QoS of all connections is deteriorated independent of source type and behaviour.

In data networks a different scheme, *rate sharing*, is usually adopted. The aim of rate sharing is to use available bandwidth to the maximum while ensuring at least some degree of fair sharing between contending users [2]. Rate sharing requires very large buffers at every network node, if the traffic process is similar to that of local area networks. Because the queue length distribution depends significantly on several traffic parameters (e.g., burst and silence length distribution) it is very difficult to make any guarantees about the mean delays and delay variation. In addition, a fair share of bandwidth remains a problem if FIFO buffers are used.

The fairness problem can be alleviated by using different schemes, e.g., virtual spacing [2], round robin/priority scheduling [3], weighted round-robin scheduling [4], dynamic time-slice [5], asynchronous time sharing [6], and per virtual circuit buffer limit [7]. Although these schemes make possible to protect the well-behaved connections from the misbehaving ones, the complexity of implementation may restrict their applicability, see e.g. [2], [8], and [9].

An approach which makes possible a simpler implementation is to offer Available Bit Rate (ABR) or Unspecified Bit Rate (UBR) service. In both services the accessible bandwidth for a connection depends on the traffic condition in the bottleneck node. In ABR service each traffic source is informed about the maximum momentary rate possible to offer by the network, whereas the UBR service counts more on the proper function of higher layer protocols if frames are lost due to overload situation in the ATM network.

In order to provide bandwidth fairness among competing ABR or UBR connections, the buffer capacity at each switch needs to be allocated fairly among the competing connections. Otherwise a connection that gets more than its fair share of the buffer space, also gets more than its fair share of the bandwidth. This is true in case of both FIFO and per VC buffer management, since the buffer space is always limited and cannot be statically allocated for each connection. In this paper we propose an allocation scheme that allows a fair share of link capacity using only a pure FIFO buffer. The fairness of the allocation scheme has been studied in some simple cases. The results are very promising although more extensive studies are needed to assess the real performance of the scheme.

2. The Buffer Allocation Scheme

The basic idea of the proposed allocation scheme is that the buffer implementation should be as simple as possible, whereas it is possible to allow a relatively complex algorithm to decide whether an incoming cell should be accepted or rejected. If this acceptance algorithm is sufficiently fair, there is no need to use more complex queue disciplines than FIFO for ABR or UBR class of service.

Let us suppose that incoming cells to an ATM node are generated by several sources, all of which send AAL5 frames. Firstly, it is important to know whether the arriving cell is the first cell of a frame, and whether some cells of the same frame have already been delivered forward. If there is an impending danger of buffer overflow, whole frames should be dropped instead of individual cells. The random early drop (RED) scheme has shown to have much better performance than a simpler scheme in which the remainder of a frame is discarded after one cell is dropped [8]. Therefore, if the buffer occupancy exceeds a certain limit, the first cell and all the following cells of a frame should be dropped. Consequently, if the first cell of a frame is accepted into the buffer, all the following cells of the frame should also be accepted, provided that the buffer is not fully occupied.

Now the first algorithm (*A1*) can be defined as follows:

- The first cell of an AAL5-frame is dropped if $X > R$, where X is the number of cells in the buffer and R is a limit for buffer occupancy.

Unfortunately, this buffer allocation scheme does not guarantee a fair share between different type of connections if some connections have exploited more resources than the other ones. If a network operator´s intention in an overload situation is to share the link capacity evenly among all active connections during an overload situation, then the operator should drop cells from those connections that have exploited the largest part of the link and buffer capacity. One way to do this is to consider the number of cells in the buffer. If a connection gets more than its fair share of the buffer space, it also gets more than its fair share of the bandwidth. Therefore, by restricting the number of cells each connection can have in the buffer one can also share the link capacity fairly.

This type of approach has been presented by Huang and Wu [10]. However, the application of the scheme differs essentially from our approach. Their system consists of two traffic classes (delay-sensitive and loss-sensitive classes) and an output-buffered ATM switch. In *arrival rate based state-dependent priority scheme* both the number of cells that a class of traffic has in the buffer and the corresponding arrival rate are used to decide which class is selected for service in the next time slot. Ramamurthy and Dighe have proposed that buffer threshold for high running VC's should be modified if cells have to be discarded because of full buffers [4]. However, they do not present any explicit method to distinguish high running connections from the other ones.

Let us denote the number of cells connection i has in the buffer by Y_i and the total number of active connections by N_a. More precisely, N_a is the number of connections which have at least one cell in the buffer. If during an overload situation a connection has more cells in the buffer than the average value (i.e., $Y_i > X/N_a$) , we may conclude that the connection is to some extent responsible for the overload situation.

Let us denote

$$W_i = \frac{Y_i N_a}{X}.$$

Parameter W_i can be used as a measure of the exploitation of network resources provided that the buffer capacity is sufficiently large in comparison with the typical burst size. In contrast, if the buffer size is small as compared of burst size, the algorithm may not offer an appropriate level of performance (see table 6 in Section 5). Now we can define an advanced algorithm by applying parameter W_i: In the second algorithm (*A2*) the first cell of an AAL5-frame is dropped if

$$(X > R) \text{ and } (W_i > 1).$$

Although this simple scheme levels down in some degree the differences in link capacity used by connections, the result is not quite satisfactory. When X exceeds R all connections which have more cells in the buffer than average (i.e., $W_i > 1$) experience roughly the same cell (or frame) loss ratio, independent of the instantaneous rate of the connection.

In order to amend the performance of the algorithm we can replace the on/off type of rejection function by a smoother one. In the third algorithm (*A3*) the first cell of an AAL5-frame will be dropped if:

$$(X > R) \text{ and } W_i > Z\left(1 + \frac{K - X}{X - R}\right)$$

where K is the buffer capacity in cells and Z is a free parameter (typically from 0.5 to 1). This formula can also be presented in a simpler form:

$$Y_i N_a(X - R) > Z(K - R)X$$

Because the term $Z(K\text{-}R)$ does not depend on the traffic condition, the implementation of Algorithm 3 actually requires 3 multiplications, one subtraction and one comparison.

The forms of the rejection functions of different algorithms are presented in Figure 1. If we apply Algorithm 3 and the buffer occupancy exceeds R, cells are rejected from a connection only if it has a considerably amount of cells in the buffer. For instance, if K = 2000, R = 1500, Z = 0.5, X = 1525, and N_a = 50, each connection is allowed to have 305 cells in the buffer. When X approaches K, the allowed number of cells decreases eventually below 1 (if Z < 1). As a consequence, finally most of the cells (= first cells of AAL5-frames) will be dropped, because the scheduling algorithm tends to level down the differences in the number of cells while X is increasing. This property is desirable since some of the buffer capacity should be left for the remaining cells of accepted frames. An appropriate behaviour of algorithm can be obtained by a proper selection of parameters R and Z.

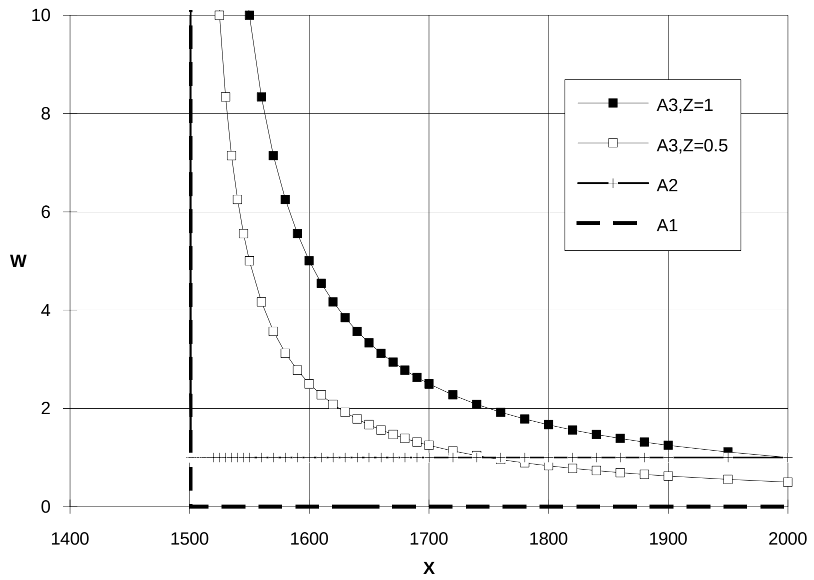

Figure 1. Rejection functions of Algorithms 1, 2 and 3

3. Fairness

In order to assess the fairness of the proposed scheduling schemes we should have a reference model. The reference model in this paper is based on the assumption that traffic variations are very slow. In this case a fair scheduling scheme rejects cells in a way that there is a maximum cell rate $w_{\max}$ which satisfies the following condition when the offered load exceeds the link capacity:

$$\sum_{i=1}^{N} w_i \left(1 - B_i'\right) = C$$

where w_i is the instantaneous cell rate of connection *i*, *C* is the link rate and:

$$B_i' = 0 \qquad \text{if } w_i \le w_{\max}$$

$$B_i' = \frac{w_i - w_{\max}}{w_i} \quad \text{if } w_i > w_{\max}$$

In other words, $w_{\max}$ is the maximum bandwidth which the connection *i* can employ during the overload situation. Parameter $w_{\max}$, and consequently B_i's, can be resolved quite easily. In addition, we should take into account that the buffer capacity reduces the actual cell loss ratio. Therefore the final reference value for cell loss ratio of connection *i* is defined as:

$$B_i^* = \frac{\sum_{j=1}^{N} m_j B_j}{\sum_{j=1}^{N} m_j B_j'} B_i'$$

where m_i is the mean bit rate of the connection and *N* is the number of connections. On the basis of the above reference model the following fairness index can be introduced:

$$F = 1 - \frac{\sqrt{\sum_{i=1}^{N} m_i \sum_{i=1}^{N} m_i \left(B_i - B_i^*\right)^2}}{\sum_{i=1}^{N} m_i B_i^*}$$

where B_i is the simulated cell loss ratio of connection *i*. This fairness index has the following properties:

- *F*=1 if $B_i = B_i^*$ for all connections.
- If $m_i = m_0$, $B_i^* = B_0^*$ for every *i*, and B_i s are exponentially distributed random variables, *F* is typically about 0.
- *F* does not change if every B_i and B_i^* is multiplied by the same constant.
- *F* is weighted by the mean bandwidth of each connection in order to avoid the overweighting of connections with very small cell rate.

4. Simulation results

In this section we present some preliminary results concerning the fairness of algorithms *A1*, *A2* and *A3*. The results are based on simulations with only one ATM-node. Although this limitation restricts the applicability of the results, the main properties of the allocation scheme can presumably be obtained.

The aim of the algorithms is to share the link capacity fairly during an overload situation. Therefore the simulations have been performed in a way that the average offered load is nearly 1, and because of traffic variations the load exceeds the link capacity almost half of the simulation time. As a consequence, the average cell loss ratios observed in the simulations represent the cell loss ratios during overload situation rather than the average cell loss ratios during long periods. The source types used in simulations are presented in Table 1.

Source	on-probability*	peak rate/ link rate	frame size in cells	mean rate/ link rate
S1	1	0.01	10	0.01
S2	0.5	0.08	4	0.04
S3	0.5	0.12	6	0.06
S4	0.5	0.16	8	0.08
S5	0.5	0.20	10	0.10
S6	0.2	0.50	10	0.10

* in the simulation program the activity of a source is determined randomly at every 8000 time slots

Table 1. Source types

In all cases the following algorithms have been compared:

- Pure FIFO queue
- Algorithm 1, R=1500
- Algorithm 2, R=1500
- Algorithm 3, R=1500, Z=0.5, 0.9 and 1

In the first simulation series (Table 2) we have tried to appraise the ability of different algorithms to discern connections with moderately different rates. Firstly, we can see that Algorithm 1 has a notable impact on the perceived cell loss ratio due to its principle that only whole frames are dropped (note that all cells belonging to a frame are supposed to be lost if at least one of the cells is dropped). However, the fairness of algorithm 1 is poor, since all connections encounter almost the same cell loss probability—even a FIFO queue is better in this respect.

Algorithm 2 offers much better fairness although it cannot distinguish very well small differences in mean rates. In contrast, the results of algorithm 3 are quite satisfactory: fairness index is as high as 0.96 and the difference of cell loss ratios between the reference model and simulation result is small especially with source types *S3*, *S4* and *S5*. With *S2* there is a larger relative error, but because of the form of fairness index these errors have an insignificant effect on the overall fairness.

sources / Algorithm	5**S2*		4**S3*		3**S4*		3**S5*			
	B_i simul.	B_i^* refer.	B_i simul.	B_i^* refer.	B_i simul.	B_i^* refer.	B_i simul.	B_i^* refer.	B_{ave}	fairness index
Pure FIFO	0.0443	2.38E-4	0.0734	0.0270	0.0860	0.1039	0.1307	0.1829	0.0881	0.51
A1	0.0647	2.28E-4	0.0667	0.0209	0.0639	0.0763	0.0640	0.1336	0.0648	0.17
A2	5.28E-4	2.30E-4	0.0362	0.0199	0.0827	0.0741	0.1112	0.1313	0.0632	0.77
A3, Z=0.5	4.00E-4	2.12E-4	0.0233	0.0193	0.0731	0.0716	0.1211	0.1257	0.0608	0.95
A3, Z=0.9	2.80E-4	1.78E-4	0.0215	0.0185	0.0689	0.0679	0.1157	0.1191	0.0576	0.96
A3, Z=1	8.57E-4	1.81E-4	0.0217	0.0191	0.0724	0.0714	0.1217	0.1250	0.0605	0.96

Table 2. Simulation results with sources *S2*, *S3*, *S4* and *S5*

In the next simulation cases (Table 3) there are sources with essentially smaller mean rate in addition to the sources with high bandwidth demand. The low rate sources could be either CBR sources or, more likely, sources that have both a high activity level and a low activity level (e.g., file transfer with user responses). The effect of the sources with low instantaneous traffic demand is presumably rather small on the offered load, but they may increase considerably the number of active sources (i.e., parameter N_a). The effect of these sources is discernible with Algorithm 2, which yields almost the same cell loss ratio for sources *S3*, *S4* and *S5*, although the reference model gives clearly different cell loss probabilities. Again, the results are much better with Algorithm 3. An especially important property of Algorithm 3 is that low bit rate sources (*S1*) did not experience any cell losses during any simulation.

sources:	20**S1*		4**S3*		3**S4*		3**S5*			
Algorithm	B_i simul.	B_i^* refer.	B_i simul.	B_i^* refer.	B_i simul.	B_i^* refer.	B_i simul.	B_i^* refer.	B_{ave}	fairness index
Pure FIFO	0.0895	0	0.0630	0.0241	0.0784	0.0989	0.1024	0.1767	0.0842	0.27
A1	0.0483	0	0.0619	0.0156	0.0619	0.0683	0.0621	0.1254	0.0589	0.20
A2	0	0	0.0627	0.0138	0.0730	0.0640	0.0710	0.1173	0.0548	0.34
A3, *Z*=0.5	0	0	0.0223	0.0160	0.0703	0.0683	0.1180	0.1247	0.0589	0.92
A3, *Z*=0.9	0	0	0.0195	0.0153	0.0672	0.0658	0.1153	0.1197	0.0565	0.94
A3, *Z*=1	0	0	0.0191	0.0149	0.0657	0.0645	0.1133	0.1176	0.0556	0.94

Table 3. Simulation results with sources *S1*, *S3*, *S4* and *S5*

In the last cases we examine the effect of connections with very high cell rates. Two sources of type 6 generate cells with half of the link rate during active periods. As regards the fairness of allocation scheme, it is important that these sources experience essentially higher cell loss ratio than the other connections. Algorithms 2 and 3 satisfy this requirement whereas the performance of *A1* is deficient.

sources:	2**S6*		4**S3*		3**S4*		3**S5*			
Algorithm	B_i simul.	B_i^* refer.	B_i simul.	B_i^* refer.	B_i simul.	B_i^* refer.	B_i simul.	B_i^* refer.	B_{ave}	fairness index
Pure FIFO	0.2716	0.5108	0.0884	5.67E-3	0.1086	0.0437	0.1523	0.1109	0.1503	0.19
A1	0.1439	0.3521	0.0846	3.60E-3	0.0956	0.0310	0.0991	0.0768	0.1039	-0.04
A2	0.3502	0.3718	5.68E-3	3.53E-3	0.0433	0.0329	0.0842	0.0803	0.1094	0.90
A3, *Z*=0.5	0.3574	0.3509	4.67E-3	3.47E-3	0.0322	0.0314	0.0702	0.0762	0.1035	0.96
A3, *Z*=0.9	0.3516	0.3433	4.17E-3	3.15E-3	0.0293	0.0288	0.0661	0.0729	0.1002	0.95
A3, *Z*=1	0.3509	0.3487	6.84E-3	3.39E-3	0.0312	0.0301	0.0698	0.0749	0.1023	0.97

Table 4. Simulation results with sources *S3*, *S4*, *S5* and *S6*

According to the above tables factor *Z* does not have any significant effect on the fairness index and, in consequence, it is perhaps not a necessary parameter. However, because it does not complicate the implementation of the algorithm, it can be applied to regulate the behaviour of the allocation process. In particular, if the cell rate of most of the connections is roughly the same and *Z*=1, the probability that the buffer will be fully occupied is relatively large. In this case some frames might be lost partially, which leads to an inefficient use of network resources. The actual selection of parameters R (in relation to buffer capacity) and Z depends on the properties of real traffic in ATM networks.

5. Implementation aspects

Although the formula used in the proposed algorithm is relatively simple, Algorithm 3 requires 3 multiplications. Because the constraint for calculation time is very stringent at high cell rates, an implementation, in which multiplications are avoided by using beforehand calculated tables, could be more expedient. A prime issue as regards the applicability of this approach is the required amount of memory. The table could be realised in a way that it contains the allowed value for Y_i (the number of cells of the connection in the buffer) for each value of X (the number of cells in the buffer) and N_a (the number of active connections). If no compression is used the size of the table is (K-R)*K*2 bytes (note that the maximum number of *active* connections is K because of the definition applied). This is presumably too large a table for practical implementations.

Fortunately, it is possible to reduce the actual size of both terms, K-R and K. In practical cases it is very unlikely that all cells in the buffer belong to different connections. Consequently, a maximum value for the number of active sources used in the table could be essentially smaller than K, and if the actual number of active sources exceeds the limit, the largest value in the table is used.

Moreover, as the results presented in the tables 2, 3 and 4 have shown, the actual form of rejection function may vary considerably without any significant effect on the performance of the algorithm (the change of fairness index is quite small between cases Z = 0.5 and Z=1). Therefore, it is not necessary to have every value of X in the table. We have performed some simulations with different granularities (G) by replacing X in the decision formula by a rounded one:

$X^* = G\lfloor X/G + 0.5 \rfloor$, where $\lfloor\ \rfloor$ denotes the integer part.

The results presented in table 5 show that we can use rather coarse granularity: even 5 different levels for the allowed W_i is in some cases sufficient. By applying conservative approximations: 1000 for the number of active connections and 50 for the number of W_i levels, we need a table of size 100 kbytes (note that the granularity depends on the buffer size). This size is most likely acceptable in real implementations.

Granularity (G):	1	10	25	50	100
Number of levels:	500	50	20	10	5
Sources					
5*$S2$ + 4*$S3$ + 3*$S4$ + 3*$S5$	0.95	0.95	0.94	0.92	0.88
20*$S1$ + 4*$S3$ + 3*$S4$ + 3*$S5$	0.92	0.88	0.83	0.79	0.70
2*$S6$ + 4*$S3$ + 3*$S4$ + 3*$S5$	0.96	0.96	0.96	0.97	0.92

Table 5. The effect of granularity on fairness index (K=2000, R=1500, Z=0.5)

Another problem of implementation is that in order to guarantee a sufficient fairness the buffer size should be quite large compared with the frame size. It is quite easy to notice that if the idle time between successive frames of a connection is longer than the emptying time of the buffer, a new frame will always be accepted. Thus, it is advantageous for the customer to send large frames with as high rate as possible. The primary solution to this problem is to use large buffers. In our simulation cases the buffer size is only 2000 cells because of restrictions in simulation program but in practical implementations the buffer size could be much larger. The essential factor is the ratio of buffer size to frame size. We have performed some simulations with varying frame sizes. The traffic situations are similar to those used in the above table expect that the sources *S5* have different frame sizes (mean and peak rates of the sources remain unchangeable). By increasing the frame size sources could reduce their frame loss rations, and in consequence, the frame loss ratio of other sources will be increased. On the basis of simulation (see Table 6) the ratio of buffer size to maximum frame size should be about 50, which means that the buffer size should be about 10000 cells if the maximum frame size is about 200 cells.

The frame size of sources S5:	10	20	32	50	100
Ratio of buffer size to maximum frame size:	200	100	62.5	40	20
5**S2* + 4**S3* + 3**S4* + 3**S5*	0.95	0.92	0.89	0.87	0.78
20**S1* + 4**S3* + 3**S4* + 3**S5*	0.92	0.88	0.84	0.77	0.66
2**S6* + 4**S3* + 3**S4* + 3**S5*	0.96	0.96	0.95	0.95	0.90

Table 6. The effect of frame size on fairness index (*K*=2000, *R*=1500, *Z*=0.5)

6. Modifications

As noticed the previous section, if the buffer size is not sufficiently large in comparison with frame size, fairness could be deteriorated. One possible solution is to perform the checking procedure when the last cell of a frame arrives at the buffer. The algorithm 3 can be preserved unchanged. If the result of Algorithm 3 is rejection and the buffer occupancy exceeds the limit *R* when first cell of next frame comes, the frame will be dropped. After this modification the frame loss ratio is the higher the larger is the frame size (as opposed to the original algorithm). On the network performance point of view this property could be desirable.

Algorithm 3 is a suitable scheme provided that all connections are supposed to have the same bandwidth requirements. However, in practical implementations there may arise a need for allocating different bandwidth for different connections. This type of allocation scheme can be realised by using weighting coefficients in algorithm 3. Let us suppose that every connection has a weighting coefficient (q_i), which determines the share provided for the connection during an overload situation. By using these coefficients the parameter W_i can be modified in the following way:

$$W_i = \frac{Y_i Q}{X q_i}$$

where Q is the sum of weighting coefficients of active connections:

$$Q = \sum_{j=1}^{N_a} q_j$$

The decision formula is similar to that of Algorithm 3: in Algorithm 4 the first cell of an AAL5-frame will be dropped if:

$$Y_i Q(X - R) > Z(K - R) q_i X$$

It should be noted that Algorithm 4 can be reduced to Algorithm 3 by using equal weighting coefficients.

7. Conclusions

This paper shows that it is possible to attain a high fairness by using FIFO buffers and a simple allocation scheme. The proposed allocation algorithms are especially suitable for the allocation of UBR connections, because it reacts accurately and quickly during overload situation and reduces the bandwidth only by those connections that use an excessive amount of the link bandwidth. At the same time, connections with small cell rate do not experience any cell losses. By using weighting coefficients it is possible to allocate link capacity according to pre-defined values. However, further simulations with several nodes and upper protocol levels are needed to assess the real performance of the buffer allocation schemes.

REFERENCES

[1] International Telecommunication Union, Telecommunication Standardization Sector (ITU-T), Recommendation I.371, Traffic Control and Congestion Control in B-ISDN, Geneva 1993.

[2] Roberts, J ‘Rate envelope multiplexing and rate sharing in B-ISDN’, To be appeared in *IEICE Transactions on Communications (Japan),* 1995.

[3] Chen, W-T and Liu, U-J ‘A feasible framework of traffic control on an ATM wide-area network’, *Computer Networks and ISDN Systems*, Vol 27 (1994) pp 67-84

[4] Ramamurthy, G and Dighe, R S ‘A multidimensional framework for congestion control in B-ISDN’, *IEEE J Selected Areas in Commun.*, Vol 9 No 9 (December 1991) pp 1440-1451

[5] Sriram, K ‘Methodologies for Bandwidth Allocation, Transmission Scheduling, and Congestion Avoidance in Broadband ATM Networks’, *Computer Networks and ISDN Systems*, Vol 26 (1993) pp 43-69

[6] Hyman, J M, Lazar, A A and Pacifici, G ‘A Separation Principle Between Scheduling and Admission Control for Broadband Switching’, *IEEE J. Selected Areas Commun.*, Vol 11 No 4 (1993) pp 605-616

[7] Doshi, B T and Johri, P K ‘Communication protocols for high speed packet networks’, *Computer Networks and ISDN Systems*, Vol 24 (1992) pp 243-273

[8] Newman, P ‘Traffic Management for Local Arena Networks’, *IEEE Communications Mag.*, Vol 32 (August 1994) pp 44-50

[9] Saunders, S ‘ATM forum ponders congestion control options’, *Data Communications*, (March 1994) pp 55-60

[10] Huang, T-Y and Wu, J-L ‘Performance analysis of prioritized state-dependent buffer-management schemes in ATM networks’, *Computer Networks and ISDN Systems*, Vol 27 (1994) pp 45-66