[7] R. Jain, S. Kalyanaraman, R.Goyal, S. Fahmy, and R. Viswanathan "The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks," *submitted to IEEE/ACM Transactions on Networking*, January 1997.

[8] A. W. Barnhart,"An Example switch algorithm for section 5.4 of TM spec.," *ATM Forum/95-0195*, February 1995.

[9] L. Kalampoukas, A. Varma, and K.K Ramakrishnan,"An efficient rate allocation algorithm for ATM networks providing max-min fairness," Technical Report UCSC-CRL-95-29, Computer Engineering Department, University of California, Santa Cruz, June 1995.

[10] N. Golmie, Y. Chang, D. Su,"NIST ER Switch Mechanism (An Example)," *ATM Forum/95-0695*, June 1995.

[11] Y. Chang, N. Golmie, L. Benmohamed, D. Su, "Simulation Study of the New Rate-based EPRCA traffic management mechanism," *ATM Forum/94-809*, September 1994.

[12] K-Y. Siu, H-Y. Tzeng, "Adaptive Proportional Rate Control (APRC)," *ATM Forum/94-0888*, September 1994.

[13] J. C. Bennett and G. T. des Jardins, "Comments on the July PRCA Rate Control Baseline," *ATM Forum/94-0682*, July 1994.

[14] Y. Chang, N. Golmie, D. Su, "Study of Interoperability between EFCI and ER Switch Mechanisms for ABR Traffic in an ATM Networks," *Computer Communications*, Vol. 16, pp. 653-658, October 1996.

[15] R. Simcoe, "Test Configurations for Fairness and Other Tests," *ATM Forum/94-0557*, July 1994.

[16] Y. Afek, Y. Mansour, and Z. Ostfeld, "Phantom: A simple and effective flow control scheme," Proceedings of the ACM SIGCOMM'96, p. 169-182, August 1996.

[17] A. Charny, K.K. Ramakrishnan,"Time Scale Analysis of Explicit Rate Allocation in ATM Networks," Proceedings of IEEE Infocom'96, Vol. 3, pp. 1182-1189, March 1996.

[18] N. Golmie, A. Koenig, and D. Su, "The NIST ATM Network Simulator, Operation and Programming," Version 3.0, *National Institute of Standards and Technology*, Internal Report 5703R1, March 1998.

[19] V. Paxson, "Fast Approximation of Self-Similar Network Traffic," Laurence Berkeley Laboratory, University of California, Berkeley, CA, LBL-36750, April 1995.

[20] R. Jain, "Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey," *Computer Networks and ISDN Systems*, Vol. 28, no. 13, October 1996, pp. 1723-1738.

[21] Y-C. Lai, Y-D. Lin, "Interoperability of EFCI and ER Switches for ABR Services in ATM Networks," IEEE Networks Magazine, p. 34-42, January/February 1998.

[22] N. Plotkin, J. Sydir,"The Rate Mismatch Problem in Heterogeneous ABR Flow Control," Proceedings of IEEE INFOCOM'97, Vol. 3, p. 1304-1314, April 1997.

of ERICA+ and Phantom outperforms both cases where all switches are ERICA+ (Figure 5(a)) and all switches are Phantom (Figure 4(a)). The ACR rates for all connections converge and stabilize at the max-min fair share. The oscillatory behavior exhibited in Figure 4(a) has disappeared. Further study of the ACR control shows that SW6 represents the bottleneck for connections B. In Figure 4 we notice a dip in the ACR for connections B between $120 - 130$ms. This causes an aggressive ACR increase for connections A that share links BB1, BB2 and BB3 with connections B and leads to rate instability and oscillation cycles. However, in Figure 9(b) where ERICA+ is used for SW6, the ACR for connections B are maintained at the max-min fair share. This eliminates the sharp increase for connections A and stabilizes the ACR of all connections. Even when SW6 is replaced with NIST ER which is more oscillatory than ERICA+ and Phantom we still see a flattening of the ACR rates (Figure 9(c)). The rate control of the B connections has a ripple effect on all the other VCs.

5 Concluding Remarks

In this paper we give an overview of some design issues for ABR switch mechanisms. We discuss different feedback types (ER and EFCI), congestion criteria and queuing disciplines. Several ER algorithms that already appeared in the literature are also given as examples. An evaluation of their performance is presented for the Generic Fairness Configuration (2) with persistent and VBR self-similar traffic. Our results show that various ER techniques can achieve fairness of bandwidth allocation using different design objectives such as minimizing buffer occupancy and maximizing link utilization. Finally we show that combining different ER flow control algorithms may in some cases result in improved performance.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, Inc. 1992.

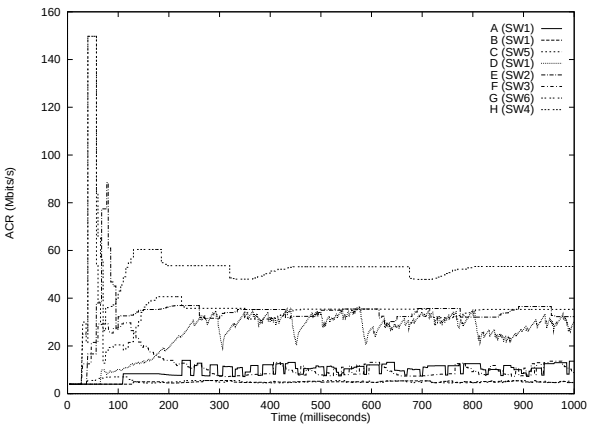
[2] "ATM Forum Traffic Management Specification Version 4.0", April 1996.

[3] R. Jain, K.K. Ramakrishnan and D.M. Chiu, "Congestion avoidance in computer networks with a connectionless network layer," Tech. Report, DEC-TRI-506, Digital Equipment Corporation, August 1987.

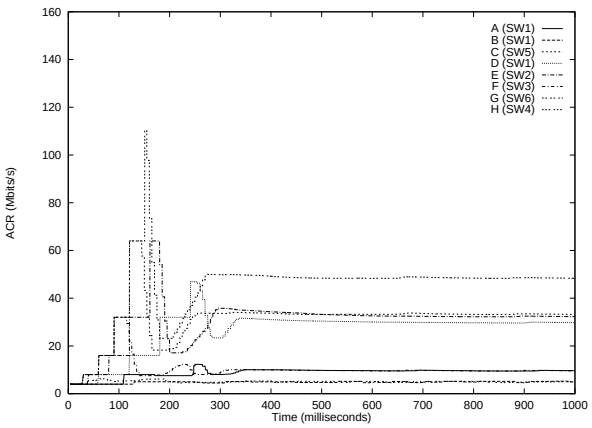
[4] A. Charny, "An algorithm for rate allocation in a packet-switching network with feedback," M.S. Thesis, Laboratory of Computer Science, MIT, Technical Report MIT/LCS/TR-601, April 1994.

[5] L. Roberts, Enhanced PRCA (Proportional Rate-Control Algorithm), *ATM Forum/94-0735R1*, August 1994.

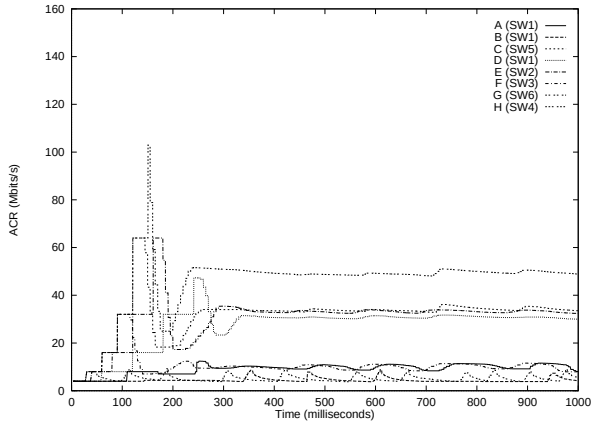
[6] K.Y. Siu, H.Y. Tzeng, "Intelligent Congestion Control for ABR Service in ATM Networks", *ACM SIGCOMM, Computer Communication Review*, pages 81-106, Volume 24, Number 5, October 1995.



(a) SW1 = NIST, All ERICA



(b) SW6 = ERICA, All Phantom

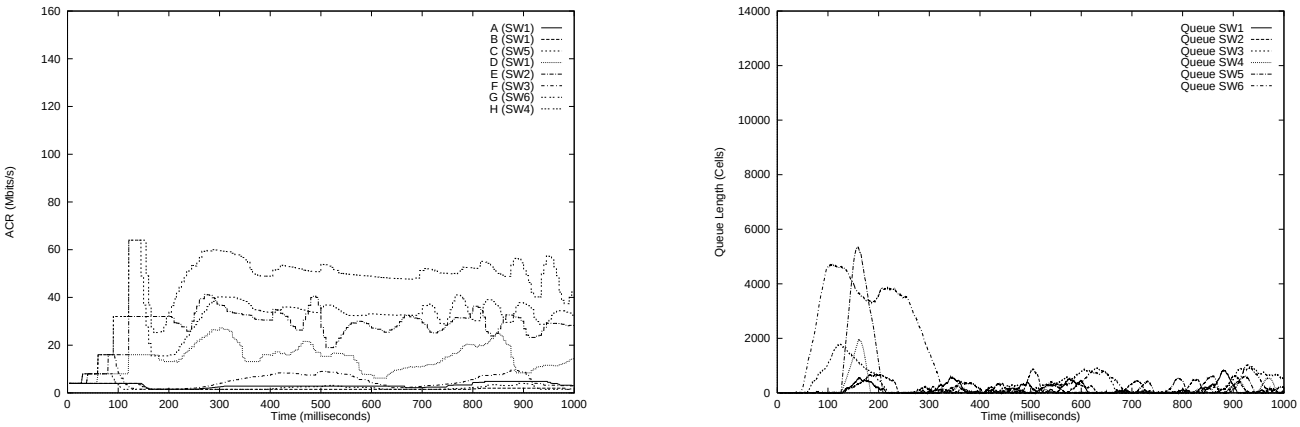


(c) SW6 = NIST ER, All Phantom

Figure 9: Experiment 3- Heterogeneous ER Environment

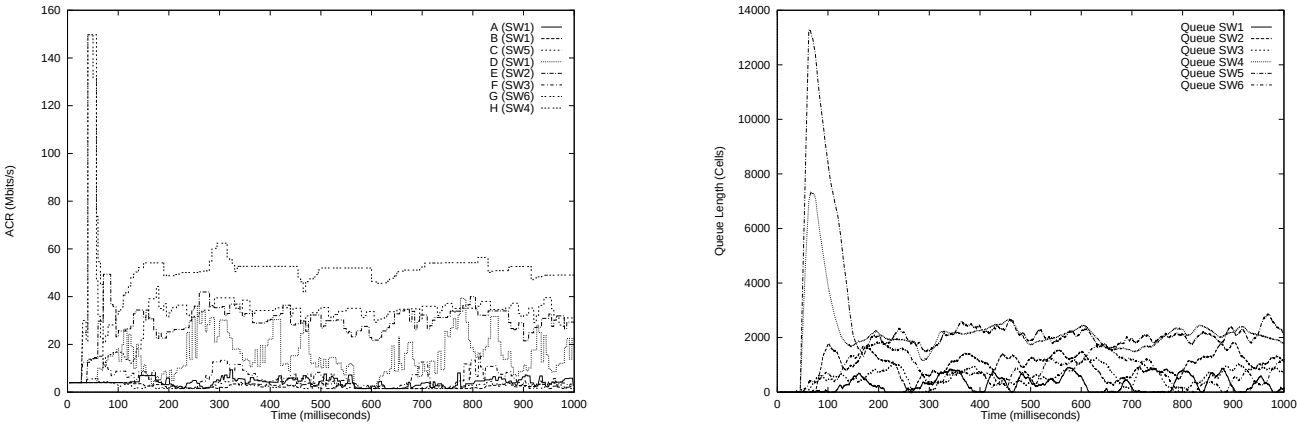
In this experiment we are concerned with the interoperability between various ER algorithms since it is very likely that switches from different vendors and thus using different ER feedback may be used in the same (WAN) topology. The impact of such intereoperation on performance is somewhat hard to quantify since it is very much dependent on the type of algorithm, the network topology and the parameters used. The interoperability issue between EFCI and ER feedback has been studied in [14] [21]. Also, the rate mismatch problem in a heterogeneous ER environment has been identified in [22]. In this experiment, we use a different algorithm for one switch of GFC-2 while keeping all the others switch algorithms unchanged. We ran a series of tests using different algorithms for different switch locations. We only present here our most interesting findings. In Figure 9(a), SW1 implements NIST ER while all the others are set to ERICA+. In this case we are able to verify the rate mismatch claim [22] that an oscillatory algorithm used at the bottleneck leads to rate oscillations and possibly unfairness. The authors in [22] show that 'proper behavior' is achieved when the least oscillatory algorithm is used at the bottleneck. In our experiment, SW1 which constitutes the bottleneck for connection D(1) interferes with its flow causing its ACR to fluctuate. In Figure 9(b) we use ERICA+ for SW6 and Phantom for the remaining switches (with $k = 5$). This combination

are not much of a concern since there is no long-term expected fair share to be reached but rather a fast and continuous adjustment of the rate in order to best utilize the resources. For this experiment, we choose ERICA+ and Phantom (with k=5) whose ACR are relatively stable in Experiment 1. DERA is not used here because our implementation requires further refinements in order to account for VBR traffic. Figure 7(a) and 8(b) show the ACR fluctuations for Phantom and ERICA+ respectively. Although the ACR oscillations are different, both schemes seem to adapt the ABR rates according to the VBR traffic. This becomes clear from the inspection of the buffer dynamics. The buffer occupancy plot for Phantom (Figure 7(b)), exhibits a similar behavior to Figure 4 indicating that the algorithm is able to adjust the ABR rates in the presence of VBR traffic. Similarly we observe no significant differences in the queuing delays between Figure 8(b) and Figure 5 which suggests that ERICA+ is also responsive to VBR traffic.



(a) ACR Dynamics (b) Buffer Dynamics

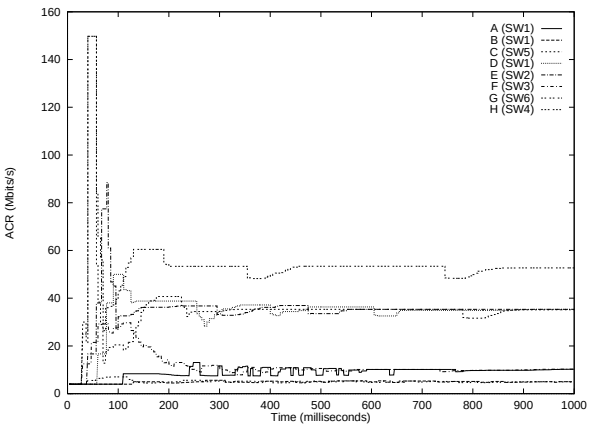
Figure 7: Experiment 2- Phantom, k=5



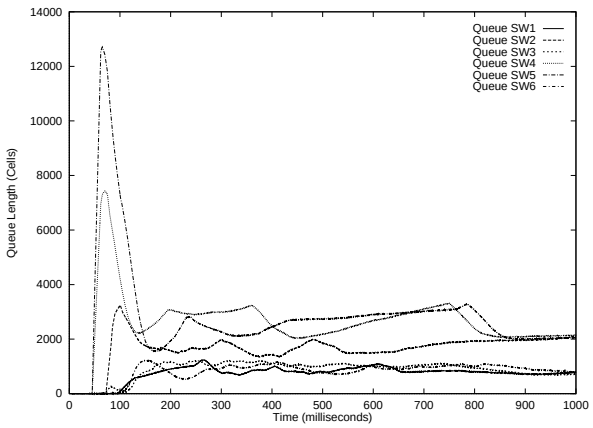
(a) ACR Dynamics. (b) Buffer Dynamics

Figure 8: Experiment 2- ERICA+

Experiment 3: Interoperability in a Heterogeneous ER Environment

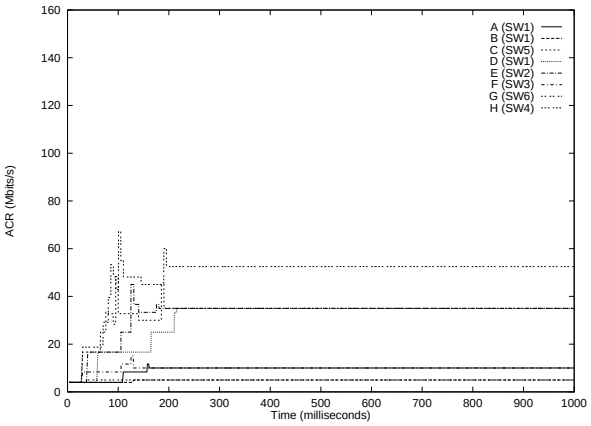


(a) ACR Dynamics.

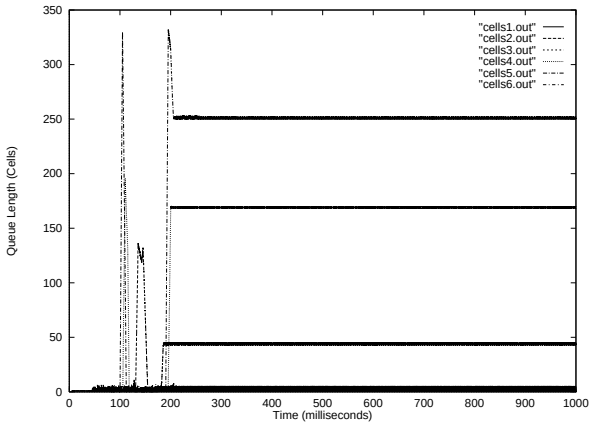


(b) Buffer Dynamics

Figure 5: Experiment 1- ERICA+



(a) ACR Dynamics.



(b) Buffer Dynamics

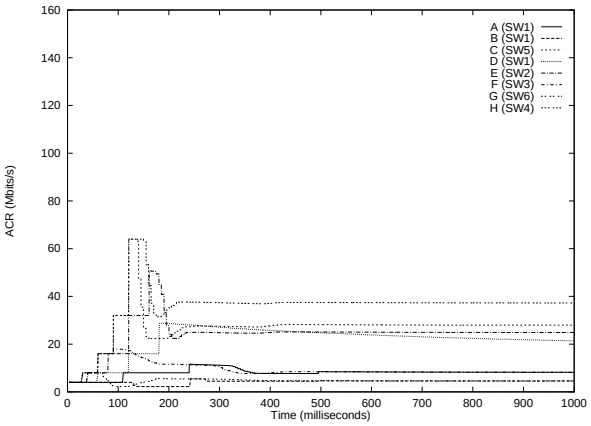
Figure 6: Experiment 1- DERA

(Figure 5(a)). The minor rate oscillations observed thereafter are a direct result of the attempt to keep the bottleneck queuing delay constant after the initial peaks (Figure 5(b)). Depending on the parameters used the buffer occupancy can get quite large. In this case we observe an initial buffer build-up of $\sim$ 13000 cells in SW6. This seems to be the price to pay in order to achieve 100% link utilization and rate stability. Figure 6 gives the results of our implementation of a DERA algorithm. The accuracy of the ER calculation causes the ACR to converge to exactly the max-min fair share after an initial ramp-up time of $\sim$ 200ms (Figure 6(a)). The buffer occupancy (Figure 6(b)) is almost negligible compared to the other algorithms. Note that the number of cells that accumulate in the buffer while the sources are adjusting their rate will remain constant after the sources have reached their fair share since no mechanism is put in place to drain the buffer.

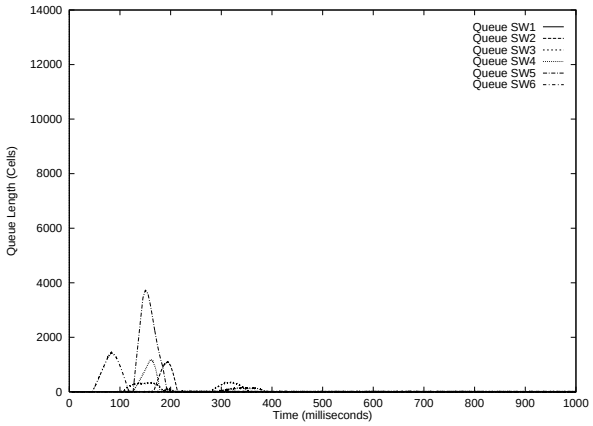
Experiment 2: Responsiveness to VBR Traffic

Our objective in this experiment is to study the ABR flow control responsiveness to changing network conditions, namely in the presence of bursty VBR traffic. In this case, convergence and rate stability

Fairness among VCs of same group is achieved although due to the lack of space we only show the rate of one VC from each group ("VC group (Entry Switch)").

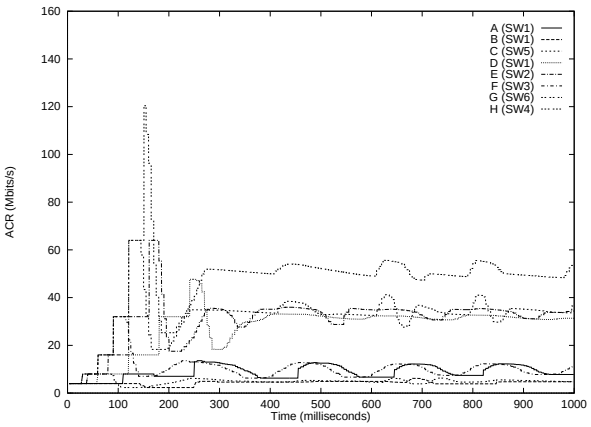


(a) ACR Dynamics.

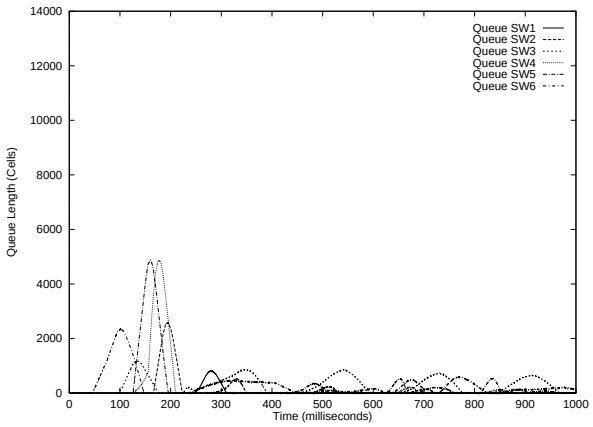


(b) Buffer Dynamics

Figure 3: Experiment 1- Phantom, k=1



(a) ACR Dynamics.



(b) Buffer Dynamics

Figure 4: Experiment 1- Phantom, k=5

In Figure 3(a), the ACR for Phantom with $k = 1$ stabilizes after $\sim$ 350ms at the phantom fair share which is slightly less than max-min fair share: 8.18, 4.54, 27.96, 18.64, 24.85 8.18, 4.54, 37.28 (in Mbits/s) for VCs A, B, C, D, E, F, G, and H respectively. Also the queue length drops to 0 after the initial transient time (Figure 3(b)). There is clearly a trade-off between the high rate stability achieved and the link utilization. In Figure 4 the utilization factor is increased to $k = 5$. This causes the ACR to oscillate in cycles of $\sim$ 200ms around the new phantom fair share computed with $k = 5$: 9.57, 4.90, 33.31, 29.61, 32.30, 9.57, 4.90, and 48.45.28 (in Mbits/s) for VCs A, B, C, D, E, F, G, and H respectively. Although the links are still slightly underutilized in Figure 4 since the phantom fair share of the VCs is still less than the max-min fair share we observe small queue build-ups in Figure 4(b).

For ERICA+, the ACR converge to the max-min fair share after a transient time of $\sim$ 200ms

ERICA+ we use $\delta = 0.1$, $T0 = 6ms$, $a = 1.15$, $b = 1$, $QDLF = 0.5$, $DecayFactor = 0.9$, $\alpha = 0.8$ and $AveragingInterval = 5ms$. For Phantom we use $\tau = 5ms$, $\alpha = 1/16$, $\beta = 1/8$, $QT = 50$ cells, $DecreasingFactor = 3/4$, $h = 1/16$, $k = 1$ or 5 and $InitPortion = 0.1$.

Parameter	Definition	Value
NRM	Number of RM cells	32
ICR	Initial cell rate value for ACR	4 Mbits/s
PCR	Peak cell rate that ACR can be set to	149 Mbits/s
MCR	Minimum cell rate for ACR	1.49 Mbits/s
RIF	Rate Increase Factor	1
RDF	Rate Decrease Factor	1/16
CRM	Missing RM cell Count	1500 cells
ADTF	ACR Decrease Factor	500 ms
CDF	Cutoff Decrease Factor	1/16
TRM	Time between forward RM cells	100 ms

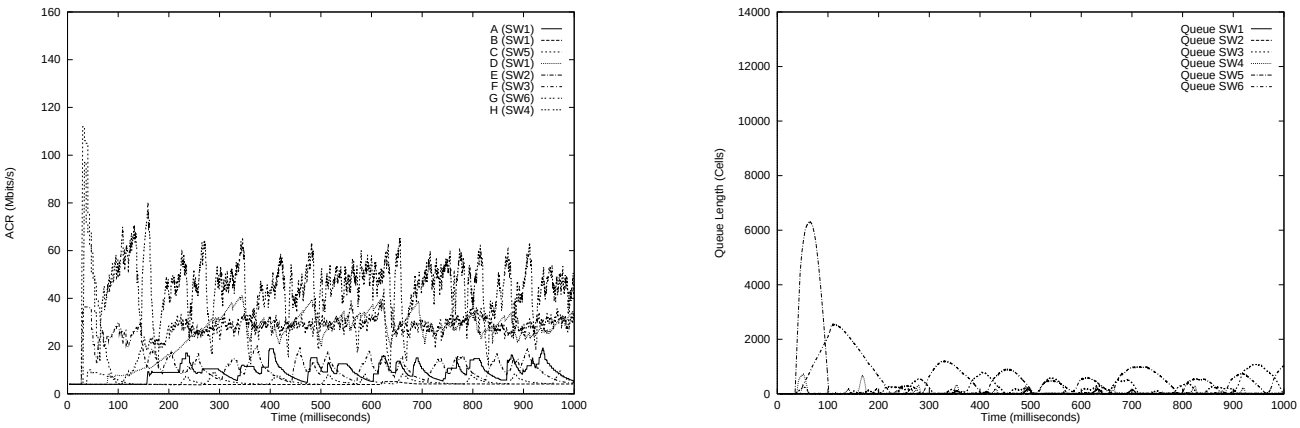
Table 1: ABR End System Parameters

4.2 Results

In this section, we present the results of three different simulation experiments that measure the performance of the ER algorithms chosen using the ACR dynamics and the buffer occupancy.

Experiment 1: Max-Min Fair Share

The primary goal in this experiment is to examine how various ER algorithms achieve max-min fairness using different design objectives and parameters. Fairness is usually measured with a Fairness Index [20]. However in this paper we use the ACR dynamics to get insights on fairness as well as convergence and stability.

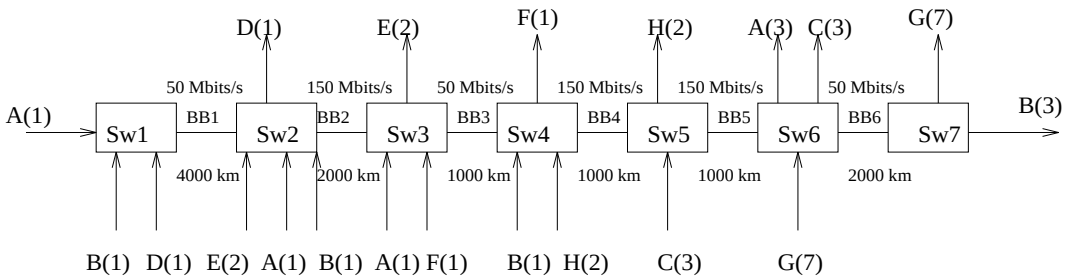


(a) ACR Dynamics.

(b) Buffer Dynamics

Figure 2: Experiment 1- NIST ER

From Figure 2(b) we note that the tight control imposed by the NIST ER on the queue length causes the ACR to oscillate around the expected fair share for the various VC groups (Figure 2(a).



Expected ACR for each VC based on a max-min allocation method:
A=F=10 Mbits/s; B=G= 5 Mbits/s; C=D=E=35 Mbits/s; H= 52.5 Mbits/s

Figure 1: Generic Fairness Configuration -2

one to create an arbitrary network topology consisting of ATM switches, ATM hosts, links between switches and hosts, and applications or traffic sources running on the hosts. The Source End System (SES) and Destination End System (DES) behavior specified in [2] were implemented on the ATM host module. The switch mechanisms NIST ER, ERICA, Phantom, and DERA described in the previous section were implemented on the ATM switch module.

Network Topology - We use the Generic Fairness Configuration (2) (GFC-2) [15] shown here in Figure 1. This configuration contains six bottleneck links supporting three types of VCs: 1) single-hop VCs (going through one link), 2) multi-hop VCs having their bottleneck link upstream and 3) multi-hop VCs having their bottleneck downstream. Some links such as link 'BB4' support all three types of VCs. Another interesting aspect of this configuration is that all multi-hop VCs are sourced at different places in the network in order to show the effect of different feedback delays and number of hops on fairness (parking lot effect). All links between the sources/destinations and switches are at a distance of 1000km and have a speed of 149.79 Mbits/s. Backbone links are set as shown in Figure 1.

Traffic Sources - Two types of traffic sources are used in the experiments. Unless specified otherwise, we use "greedy sources" such that each source generates constant bit rate traffic that fills up as much bandwidth as available to the source. In Experiment 2, we use a self-similar traffic source modeled according to [19]. The self-similar source traverses all 7 switches of the GFC-2 configuration entering at SW1 and exiting at SW7. It contributes 25 Mbits/s on average on each backbone link. We used a hurst parameter of 0.9 and a time granularity of 0.25ms for the cell generation.

Control Parameters - The values of the ABR major end system parameters used are shown in Table 1. All simulations are run for a total time period of 1000 milliseconds. All VCs of different groups are activated at simulation time 0. We use the following parameter settings for the various switch algorithms. For NIST ER, we use $QT = 250$, $AV = 1/16$, and $MRF = 0.98$. $MAIR$ was set to 0.09Mbits/s for SW1 and SW6, 0.15Mbits/s for SW2, SW3, SW5 and 0.35Mbits/s for SW4. For

computes the unused bandwidth $\Delta = AC - IR$ periodically. $BRM(ER)$ is computed according to $MACR = \max(MACR * (1 - AV) + \Delta * AV, MACR * dec_factor)$ where $0 \leq dec_factor \leq 1$ is used to limit the reduction in rate due to transient bursts. The averaging factor AV is a function of the queue length. Thus, the effective $MACR$ is controlled by the arrival rate as well as the queue occupancy. To further limit the oscillations of $MACR$, AV is made a function of the $MACR$ variance. The underutilization of the algorithm when only few sessions share a link may be reduced by using a utilization factor k. Rather than restricting the sessions by Δ they can be restricted by $k * MACR$ where $VCPS = AC/(n + 1/k)$. Unlike the algorithms discussed above where the main objective is to maximize resource utilization, the main design goal in this case is to minimize the delay and achieve fairness at the expense of underutilizing the link.

In [17] Charny describes the **Distributed Explicit Rate Allocation (DERA)** framework in which each switch ER max-min computation is based on the local information available to the switch. This local information is updated regularly upon the receipt of FRM and BRM cells and used to estimate the state of each VC. Our implementation of a DERA algorithm is similar to what is proposed in [4] and [9] with some variations. These differences are mainly: 1) no changes are required to the source end system, 2) the switch modifies the ER field of the RM cell and does not use the CCR field, 3) both FRMs and BRMs are used in updating the VC local information and updating the fair share. The algorithm operates as follows. The switch maintains a local fair share LFS computed according to FS and updated at fixed intervals. Initially, all VCs are marked unconstrained and LFS is simply the total ABR capacity divided by the number of VCs. In addition the switch maintains a rate $R(i)$ for each connection i, which is initially set to LFS. Upon receipt of a FRM, the switch compares the ER value found in the FRM, $FRM(ER)$, with LFS. If $FRM(ER) > LFS$, $FRM(ER)$ is reduced to LFS since the ER value given back to the source cannot exceed LFS. Similarly upon receipt of a BRM, $BRM(ER)$ is compared against LFS. If $BRM(ER) > LFS$, $BRM(ER)$ is reduced to $R(i)$. On the other hand if $BRM(ER) < LFS$, indicating that the VC flow is constrained somewhere else along the path, the switch saves the value of $BRM(ER)$ in $R(i)$. This quantity is used in the next update interval in order to reallocate the unused bandwidth to other VCs where all VCs with $R(i) < LFS$ are marked constrained and LFS is computed according to (1) while all unconstrained VCs $R(i)$ are set to the new value of LFS.

Four of the algorithms discussed above, namely, NIST ER, ERICA, Phantom and DERA are implemented and used in the performance evaluation presented in Section 4. The choice of the algorithms is not based on any value judgement but because they constituted different approaches for computing ER.

4 Performance Evaluation

4.1 Simulation Model

Simulation Environment - The NIST ATM Network Simulator [18] is a flexible system allowing

NIST ER presented by Golmie [10] is based on bandwidth demand estimate [2]. It borrows various features from the algorithms described above. In this case $MACR$ is used to compute an average ACR common to all VCs. The congestion is determined based on queue derivative. A positive derivative indicates an instantaneous bandwidth demand that may be exceeding the available port bandwidth in the long run and thus causes the switch to declare the state of congestion. The congestion status is set back to $FALSE$ when the queue size falls below a threshold QT insuring that during normal operation mode the switch buffer is kept small. In addition the switch monitors the aggregate queue input rate and computes LF. $LF = 1$ corresponds to the optimal case where the input rate is equal to the desired operating point or Target Rate (TR). When $LF < 1$ the aggregate input rate is less than the available bandwidth and the sources need to have their rate increased in order to better utilize the line. Hence $MACR$ is increased by the $MACR$ Additive Increase Rate (MAIR) and the explicit rate given to the sources in the BRM, BRM(ER), is equal to $MACR$. On the other hand, $LF > 1$ indicates that the line is saturated and $BRM(ER)$ is set to $MACR * MRF$ with $0 < MRF \leq 1$. Simulation results obtained with this algorithm [14] suggest that for simple WAN parking lot configurations with one type of multihop connections and several single hop connections, the ACR converge rapidly to a neighborhood of their designed operating point while the switch buffer occupancy is kept small. However with more complex configurations such as the Generic Fairness Configuration [13] where connections with several feedback delays interact, some rate instability may occur due to the algorithm's high sensitivity to queue growth.

ERICA proposed by Jain [7], controls ER based on FS, LF, and the VC share rate, VCS. LF is computed at the end of fixed intervals along with $FairShare$ which is simply the ABR capacity divided by the number of active VCs. Since some sources may not be able to increase their rate to this $FairShare$, the switch computes $VCS = CCR/LF$ in order to allocate the leftover bandwidth to sources that can use it according to the max-min fairness criteria. The CCR field is read from the FRM cell and the VCS value is increased or decreased according to the load factor changes. $BRM(ER)$ is set to the maximum of the $FairShare$ and VCS. In an enhancement to ERICA, **ERICA+**, TR is made a decreasing function of the queue length in order to better control the queuing delay. Hence if the input rate is small and the queue length is large some of the link capacity is reserved to drain the queues before the source allocation is increased. Furthermore, in order for ABR to dynamically adjust to VBR traffic, the ABR queue is kept at a target length that guarantees 100% utilization even if the VBR traffic is absent during a short interval. TR becomes equal to $f(T_q) * AC$ where T_q is the queuing delay and AC is the ABR capacity. This non-zero queuing delay represents a major difference with most ER algorithms especially those examined in this paper. The target operating point for ERICA+ is 100% utilization and a fixed non-zero queuing delay.

In **Phantom** [16] Afek suggests controlling TR by adding a "phantom" user and thus setting aside a portion of the bandwidth unused. If the number of ABR VCs is n, then the VC Phantom share, $VCPS$ is $AC/(n+1)$ where AC is the available link capacity for ABR. In the steady state operation the unused bandwidth, and the maximum share of VCs is the same as the $VCPS$ rate. The algorithm

received during ACR promotion (underload condition with $MACR < ACR$) or ACR suppression (overload condition with $MACR > ACR$).

$$MACR = MACR + (R - MACR) * AV \tag{2}$$

where R is equal to CCR of the FRM cell [5][6][10], and AV is an averaging factor controlling the sensitivity to current changes. In [8] the MACR is updated every FRM cell and $R = CCR * \min(1, TR/IR)$ where IR is equal to the input rate (per port) during one measurement interval.

Load Factor (LF) - LF is the link load factor computed at fixed time intervals such as every round trip time [7] or every N data cells [10] according to:

$$LF = \frac{IR}{TR} \tag{3}$$

3 ABR Switch Mechanisms Examples

In this section we outline samples of ABR switch algorithms that appeared in the literature. We focus on ER switch design only since the EFCI scheme is fairly simple and its performance well understood.

In **EPRCA** and **APRC** proposed by Roberts [5] and Siu [6] respectively, the switch sets the ER field in the BRM cell only when there is a congestion on the link. The ER is a function of $MACR * RF$ where RF the reduction factor can take on values between $0-1$ depending on the state of congestion. Under high congestion condition it can be set close to 0, while under moderate or no congestion it can be set closer to 1. In order to determine the congestion condition, EPRCA uses an absolute queue length threshold, while APRC uses a queue length rate change. Although these two algorithms make use of the explicit rate, their performance [11][12] is not much different from algorithms using EFCI feedback and passive congestion control. Since the explicit rate is only computed during a state of congestion no stable rate can be achieved.

In [8] Barnhart discusses **CAPC2** a form of ER control that is based on the ratio IR/TR to estimate the feedback rate. When the link is not congested, ER is increased by a fraction of the available bandwidth, that is, $(1 - IR/TR) * Gain$, where $Gain$ is set to different values for LAN and WAN networks. When the link is congested, ER is set to a fraction of $MACR$. Congestion is determined according to an absolute queue length threshold. In addition, if the queue length exceeds a certain threshold, the switch instructs the source to reduce its rate using a form of binary [1] control. The performance of CAPC2 [8] shows that queue length is kept small. In some cases this may result in a longer convergence time and larger oscillations. Also since the binary indiction bit is used in very congested states, connections going through several switches are more likely to experience the "beatdown" phenomenon associated with binary congestion feedback [13].

[1] In [8] the author suggests setting a Congestion Avoidance (CA) bit in the RM cell with an associated decrease factor at the source that is larger than RDF.

Per VC Accounting - This is a hybrid version between per VC aggregation and separation. First appearing in [7], it borrows simplicity from per port queuing and intelligence from per VC queuing. All cells are still queued per port, but the switch keeps information on each VC in look-up tables. This solution can provide better results than pure per-port queuing.

Per VC Queuing - It is clear that separating VC flows into different queues can provide not only information on each VC but also accurate control means to regulate traffic. In this case, even with EFCI feedback a Round Robin scheduler can be used to achieve fairness. Although not very popular because of the cost and complexity involved, this discipline can give close to optimal performance results. It can be used with any type of feedback and congestion criteria.

2.4 Explicit Rate Computation

The effectiveness of the ER approach depends on how a switch computes the feedback rate. The main goals are 1) to provide a fair share to all users or VCs, 2) to maximize the link utilization up to a predefined cap, and 3) to keep the buffer utilization within a reasonable range. One of the biggest challenges is to provide an algorithm that is responsive to network changing conditions but demands little processing power of the switch, as a switch may be required to support hundreds of ports and thousands of VCs. In order to compute the fair share for individual VCs, one needs to use a combination of policies in areas such as input rate computation, congestion criteria and queuing policies which are described in the previous sections. The following discusses some general factors that apply to the ER rate determination. Example ER algorithms will be given in Section 3.

Target Link Rate (TR) - TR is normally a fraction of the available link capacity for the ABR service. This fraction can be constant, for example 95% of the link capacity or it can vary as a function of the queue length for example.

Fair Share (FS) - FS is generally computed according to the Max-Min fairness criteria [1] with the assumption that all VCs have their $MCR = 0$. For a given link, the VCs can be divided into two groups: constrained and unconstrained. The constrained group are those VCs that cannot use all their fair share of the bandwidth because of a limited demand or a bottleneck somewhere else along the path. FS is then computed according to:

$$FS = \frac{AC - \sum_{i=1}^{m} C_i}{n - m} \tag{1}$$

where AC is the available ABR capacity which is the same as the target link rate TR, C_i is the constrained VC_i capacity or rate, m is the number of constrained VCs, and n is the total number of ABR VCs. Other fairness criteria with $MCR \neq 0$ suggested in [2] include techniques such as allocation proportional to MCR, MCR plus equal share, maximum of MCR or Max-Min share, and weighted allocation.

Mean ACR (MACR) - MACR is an average ACR computed at the switch and used for the estimation of ER. The $MACR$ is updated using an exponential averaging whenever a FRM cell is

2.2 Congestion Criteria

A variety of congestion control options are available to the switch in order to regulate ABR traffic. Mainly, they can be classified into two categories depending on whether they perform congestion control (passive control) or congestion avoidance (active control).

Congestion Control - In this passive mode of operation, the switch sends to the source information about its congestion only after it occurs. It does not need to monitor, measure or update other information about the traffic flow at any other time. The congestion status is declared based on either simple queue thresholding [5] or queue derivative [6]. For queue thresholding, the switch monitors its output buffer. If the number of cells in the buffer becomes greater than T_{thresh} then the switch sets its congestion status to $TRUE$. Otherwise it is set to $FALSE$. Another method consists of monitoring the queue derivative, that is the change between the level of the queue now, $QLevel_{now}$, with respect to last time checked, $QLevel_{previous}$. This method provides a tighter buffer control and thus a faster response time to congestion. If $QLevel_{now} - QLevel_{previous} > 0$, cells are accumulating at the buffer, and the congestion flag is set to $TRUE$. Once the congestion flag is set, the switch tags the EFCI bit of the data cells corresponding to either the congested port or VC, depending on the queuing policy chosen. The switch may also choose to set the CI bit of the FRM or BRM cells. Although very simple, this form of flow control leads to rate oscillations. The source rate does not converge to the fair share rate. This passive congestion control method is used mostly with binary switches.

Congestion Avoidance - Switches in this category attempt to avoid congestion. Unlike passive schemes, active switches use all the information available to them and communicate it to the source in the form of an explicit rate or fair share. An effective technique of congestion control used in the computation of the explicit rate is load monitoring where the switch monitors the traffic it receives in order to determine the bandwidth it needs to give each flow. This so-called load measurement is done on an aggregate of VCs (per port) as in [7][8]. It could also be done on a per VC basis as in [4][9].

2.3 Queuing Policies

The type of queuing discipline used in an ABR switch, is another option that is left vendor implementation dependent. This particular feature was at the heart of the debate between credit-based and rate-based both considered for ABR flow control. It was felt that the credit-based did not scale well since it required per VC queuing.

Per Port Queuing - With this queuing discipline, all VCs on the same port share the same buffer pool. Although, attractive for its simplicity, this discipline lacks fairness, especially if used in combination with EFCI marking. This is due to the fact that the switch does not differentiate between the different VCs and doesn't keep track of their individual flows. On the other hand, if used in ER switches, both fairness of bandwidth allocation and source rate convergence to fair share can be achieved.

2 ABR Switch Design

The ABR flow control mechanism provides considerable flexibility to designers in deciding ABR switch operation and resource allocation for a given end system behavior. Different switches can use different mechanisms and still intereoperate in the same network. But using different switch mechanisms leads to significant differences in terms of performance (queue length, cell loss, fairness of bandwidth allocation, throughput and link utilization), complexity and finally cost. In order to design an ABR switch mechanism one needs to understand the different flavors or options available and their impact on performance. Thus, for a given a set of requirements and evaluation criteria, the switch engineering task is made easier. We classify the ABR switches broadly into three categories based on the type of feedback, the congestion criteria and the queuing service used.

2.1 Feedback Types

Explicit Forward Congestion Indication - The EFCI mechanism originated from the DECbit scheme [3] that uses congestion avoidance in connectionless networks. Packets carry one bit in the header that can be set by any network node to indicate congestion. Based on this form of explicit congestion information the transport layer can adaptively change its window size before packets are lost. The window size is additively increased in case of no congestion and exponentially decreased in case of congestion. For ABR flow control, the window adjustments are changed to rate adjustments based on a binary congestion notification. An EFCI switch sets the EFCI bit in the data cell traveling from the source to destination when in congestion. Upon receipt of a data cell with an EFCI bit set, the destination sets the CI bit in the BRM cell before sending it back to the source. This binary feedback mechanism designed for connectionless datagram networks is rather slow and inadequate for rate-based flow control in high speed networks. Several problems are identified. Most important is the "beat-down" problem [13] where traffic passing through a large number of switches is more likely to encounter congestion somewhere along the path and thus get less bandwidth than single-hop traffic. Also, with binary feedback the sources know when to increase or decrease their rate only after some delay from the time of congestion depending on the location of the congested link, and have to guess a rate that could be supported by the network. This guessing leads to undesired oscillations and some large buffer build-ups, especially in WAN environments. Aggressive increases are always followed by sharp decreases and the cycles repeat forever. In connection oriented ATM, network nodes have more information on the flow of each VC and thus could do a better job regulating traffic.

Explicit Rate - The idea of explicit rate feedback first appeared in [4]. Currently, in the ER scheme, similar to Charny's original scheme, RM cells are sent along a connection to collect the minimum "advertised rate" from all nodes. The RM cells initially contain the source demand rate contained in the ER field and any node can reduce this field to the "advertised rate" which is the source's fair share of the available bandwidth at each node.

1 Introduction

The rate-based flow control mechanism dynamically regulates the rate of data flow into the network based on feedback information about the network state. The objective of this congestion control mechanism is to maximize throughput and minimize delays and losses incurred at intermediate bottleneck network nodes. In the case of ABR traffic in ATM networks, the leftover bandwidth is redistributed fairly (for example according to the max-min fairness criteria [1]) by the network to contending ABR sources. Each one of these sources is limited to sending its data at an Allowed Cell Rate (ACR), which is a rate given by the network. The rate-based flow control algorithm for ABR traffic in ATM networks [2] operates as follows: the source sends to the destination every NRM data cells a "forward" Resource Management (FRM) cell. This FRM cell contains several fields. In here we will only mention those fields relevant to our study; namely, the ACR field, the Congestion Indication (CI) field, and the Explicit Rate (ER) field. Upon receipt of the FRM cell, the destination returns it to the source with the latest network information. This FRM becomes a "backward" RM (BRM) cell. The network information is contained in either one or both of its CI and ER fields depending on the mode of operation of the intermediate network nodes.

Although no complete specifications are offered in [2], there are mainly two modes of operation for the network intermediate switches supporting ABR traffic: Explicit Forward Congestion Indication (EFCI) or Explicit Rate (ER). The EFCI switch has a binary mode of operation that upon congestion sets the congestion bit in either or both the data cell and the BRM cell. On the other hand, an ER switch computes an average fair share of its capacity per active Virtual Circuit (VC) flow. This quantity is called Explicit Rate (ER) and is given directly to the source in the ER field of either the FRM or BRM. The source adjusts its ACR based on the feedback information contained in the BRM cells. If the CI bit is set, the source decreases its ACR multiplicatively by the Rate Decrease Factor (RDF). The ACR may go down to Minimum Cell Rate (MCR) which is a rate negotiated at connection set up. Otherwise the source increases its ACR additively according to the Rate Increase Factor (RIF) which is a fraction of the Peak Cell Rate (PCR) and is negotiated at connection set up.

This paper is concerned with the different options for designing ABR switches, such as: feedback types, queuing disciplines, and congestion criteria. We are interested in understanding the fundamental differences between them. We try to compare their performance with respect to different network delays and traffic types.

The remainder of this paper is organized as follows. Section 2 examines several switch design options. Section 3 describes specific switch examples. In section 4 we evaluate their performance given different network conditions, such as network delays, and traffic patterns. Simulation results are given and discussed. In Section 5 we offer some concluding remarks.

ABR Switch Mechanisms: Design Issues and Performance Evaluation

N. Golmie, Y. Saintillan, and D. Su
National Institute of Standards and Technology
Gaithersburg, Maryland 20899

Abstract

Considerable protocol development efforts in recent ATM Forum activities have been centered on the traffic management of available bit rate (ABR) service. The goal is to efficiently manage the leftover network bandwidth and fairly distribute it among contending ABR virtual connections (VC) so that communication links can be optimally utilized. A rate-based flow control mechanism has been adopted for that purpose. While the ATM Forum Traffic Management Specification Version 4.0 document gives a complete description of the end system behavior, it gives only a high level description of the switch or intermediate network nodes, leaving the details of their behavior vendor implementation dependent.

This paper is concerned with the variety of ABR switch mechanisms available for implementation with the current end system definition. The various switch mechanisms can be broadly classified in two categories depending on the type of feedback information provided to the end system. The feedback mechanism is either binary involving setting the EFCI bit in the data cell header or explicit rate in which case the switch has to calculate a target rate and send it back to the source through an RM cell. Also within each category, different subgroups are formed based on the type of queuing involved (for example per port queuing, per VC accounting, per VC queuing) and the congestion declaration criteria adopted (for example queue thresholding, queue derivative, load monitoring).

Our goal is to provide an overview of the various switch mechanisms available and discuss their performance characteristics accordingly. The performance metrics used include queue length, source flow-controlled cell rate dynamics, fairness of bandwidth allocation in the steady state and ramp-up time in the network transient state. Simulations of the different switch algorithms' dynamics for network configurations and scenarios of interest are presented.

Keywords: ATM, Flow Control, Congestion Control, Traffic Management, Rate-Based Mechanism, ATM Switch Mechanisms.