Fair Network Bandwidth Allocation with Minimum Rate Guarantee and Its ABR Implementations *

Yiwei Thomas Hou† Henry H.-Y. Tzeng‡ Shivendra S. Panwar§ Vijay P. Kumar¶

Abstract

A novel concept in available bit rate (ABR) service model as defined by the ATM Forum is the minimum cell rate (MCR) bandwidth guarantee for each connection. In this paper, we present a network bandwidth allocation policy to support each ABR connection's MCR requirement, as well as its peak cell rate (PCR) constraint. Furthermore, we develop two explicit-rate (ER) based ABR algorithms consistent with the ATM Forum ABR traffic management framework to achieve this rate allocation policy. The first ABR implementation is a simple heuristic algorithm which does not require per-VC accounting. It requires minimal implementation complexity and offers satisfactory performance in a LAN environment. The second ABR implementation employs per-VC accounting and is proven to converge to our rate allocation policy for any network topology and any set of link distances.

1 Introduction

ABR service as defined by the ATM Forum supports applications that allow the source end system to adjust the information transfer rate based on the bandwidth availability in the network [1]. By the specifications in [1], on the establishment of an ABR connection, the user shall specify to the network both its MCR requirement and PCR constraint for the requested connection. The source starts to transmit at an initial cell rate (ICR), which is greater than or equal to MCR, and may adjust its rate up to PCR based on congestion and bandwidth information from the network.

A key performance issue associated with ABR service is fair allocation of network bandwidth for each virtual connection (VC). The ATM Forum has adopted the max-min fairness criterion for ABR service [2]. Prior efforts to design ABR algorithms to achieve the max-min fair rate allocation, such as [3, 7, 8, 9, 10], did not address the fairness issue in the context of each connection's MCR requirement. For connections with MCR/PCR constraints, a new definition of rate allocation is required.

We propose a rate allocation policy, called *MCRadd*, to allocate network bandwidth with MCR/PCR constraint for each ABR virtual connection. This policy was first informally introduced in [5, 13] for the simple single node case without the PCR constraint. In [4], we formally defined this rate allocation policy with MCR/PCR support. In this paper, we focus on distributed ABR implementations to achieve this rate allocation policy.

We present two distributed ABR algorithms to achieve the MCRadd policy. Both algorithms use the explicit-rate (ER) calculation. The first ABR algorithm is based on the *Intelligent Marking* technique [10, 11, 12] and does not require per-VC accounting. It is a simple heuristic algorithm and is most effective in a LAN environment. The second ABR algorithm is based on the work in [3] and requires per-VC accounting. With this additional complexity, the algorithm is proven to converge to the MCRadd policy under any network environment (LAN and WAN).

The remainder of this paper is organized as follows. Section 2 defines the MCRadd rate allocation policy. Section 3 outlines the ABR traffic management framework and defines the source and destination behavior. In section 4, we present a simple heuristic ABR switch algorithm without per-VC accounting to achieve the MCRadd policy in LAN environment. In section 5, we show a second ABR implementation employing per-VC accounting. Section 6 concludes this paper.

2 The MCRadd Rate Allocation Policy

In our model, a network $\mathcal{N}$ is characterized by a set of links $\mathcal{L}$ and sessions $\mathcal{S}$.[1] Each session $s \in \mathcal{S}$ traverses one or more links in $\mathcal{L}$ and is allocated a specific rate r_s. The (aggregate) allocated rate F_ℓ on link $\ell \in \mathcal{L}$ of the network is $F_\ell = \sum_{s \in \mathcal{S} \text{ traversing link } \ell} r_s$. Let C_ℓ be the capacity of link ℓ. A link ℓ is *saturated* if $F_\ell = C_\ell$. For feasibility, we assume throughout our paper that the sum of VCs' MCR requirements traversing any link does not exceed that link's capacity, i.e. $\sum_{\text{all } s \in \mathcal{S} \text{ traversing } \ell} \text{MCR}_s \le C_\ell$, for every $\ell \in \mathcal{L}$. This assumption is guaranteed by admission control at call setup time to determine whether or not to accept a new ABR virtual connection.

We say that a rate vector $r = (\cdots, r_s, \cdots)$ is *ABR-feasible* if the following two constraints are satisfied:

$$\text{MCR}_s \le r_s \le \text{PCR}_s \quad \text{for all } s \in \mathcal{S},$$

*Part of this work was performed while Y. T. Hou spent the summer of 1996 at Bell Labs, Lucent Technologies, Holmdel, NJ. This work is supported in part by the New York State Center for Advanced Technology in Telecommunications (CATT), Polytechnic University, Brooklyn, NY.

†Y. T. Hou is a Ph.D. candidate under the National Science Foundation Graduate Research Traineeship Program at Polytechnic University, Brooklyn, NY.

‡H. H.-Y. Tzeng is with Bell Labs, Lucent Technologies, Holmdel, NJ.

§S. S. Panwar is on the faculty of the Dept. of Electrical Engineering, Polytechnic University, Brooklyn, NY.

¶V. P. Kumar is with Bell Labs, Lucent Technologies, Holmdel, NJ.

[1] From now on, we shall use the terms "session", "virtual connection", and "connection" interchangeably throughout our paper.

$$F_\ell \leq C_\ell \quad \text{for all } \ell \in \mathcal{L}.$$

Definition 1 A rate vector r is *MCRadd fair* if it is ABR-feasible, and for each $s \in \mathcal{S}$ and every ABR-feasible rate vector $\hat{r}$ in which $\hat{r}_s > r_s$, there exists some session $t \in \mathcal{S}$ such that $r_s - \text{MCR}_s \geq r_t - \text{MCR}_t$ and $r_t > \hat{r}_t$. □

Definition 2 Given an ABR-feasible rate vector r, a link $\ell \in \mathcal{L}$ is an *MCRadd-bottleneck link* with respect to r for a session s traversing ℓ if $F_\ell = C_\ell$ and $r_s - \text{MCR}_s \geq r_t - \text{MCR}_t$ for all sessions t traversing link ℓ. □

It can be shown that the following two theorems are true [4].

Theorem 1 An ABR-feasible rate vector r is MCRadd fair if and only if each session has either an MCRadd-bottleneck link with respect to r or a rate assignment equal to its PCR. □

Theorem 2 There exists a unique rate vector that satisfies the MCRadd fair rate allocation policy. □

We construct the following centralized algorithm to compute the rate allocation for each session to satisfy the MCRadd fairness policy.

Algorithm 1 MCRadd Centralized Algorithm

1. Start the rate allocation of each session with its MCR.
2. Increase the rate of each session with the smallest rate increment such that either some link becomes saturated or some session reaches its PCR, whichever comes first.
3. Remove those sessions that either traverse saturated links or have reached their PCRs and the capacities associated with such sessions from the network.
4. If there is no session left, the algorithm terminates; otherwise, go back to Step 2 for the remaining sessions and remaining network capacity. □

As an example, for the *peer-to-peer* network configuration (Fig. 1), the output port link of SW1 (Link 12) is the only bottleneck link for all sessions. Assume that all links are of unit capacity. The MCR requirement and PCR constraint for each session are listed in Table 1. Using Algorithm 1, we obtain the rate assignment for each session in Table 1 under the MCRadd rate allocation policy.

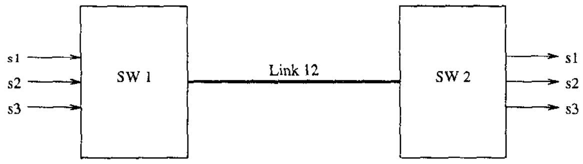

Figure 1: The peer-to-peer network configuration.

The MCRadd rate allocation policy and its centralized algorithm cannot be applied directly to a distributed ABR traffic management environment. To show the practical merit of achieving the MCRadd rate allocation policy for ABR service, we will develop two distributed ABR implementations in the following sections.

Session	MCR	PCR	MCRadd Rate Allocation
$s1$	0.15	0.30	0.30
$s2$	0.10	0.35	0.35
$s3$	0.05	0.50	0.35

Table 1: MCR requirement, PCR constraint, and MCRadd rate allocation for each session in the peer-to-peer network configuration.

3 The ABR Traffic Management Framework

A generic rate-based closed-loop flow control mechanism for ABR service is shown in Fig. 2. Resource Management (RM) cells are inserted periodically among ATM data cells to convey network congestion and available bandwidth information to the source. RM cells contain important information such as the source's allowed cell rate (ACR) (called the current cell rate (CCR) in the RM cell's field), MCR requirement, explicit rate (ER), congestion indication (CI) bit and no increase (NI) bit. A transit node and destination may set the ER field, CI and NI bits in RM cells. All RM cells of an ABR virtual connection are turned back towards its source after arriving at the destination. Upon receiving backward RM cells, the source adjusts its cell generating rate accordingly.

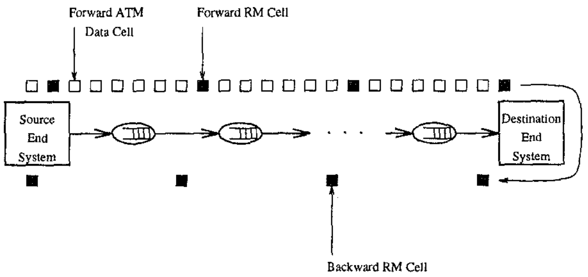

Figure 2: Closed-loop rate-based flow control for an ABR virtual connection.

The following algorithm specifies the source behavior of our ABR algorithm [1].

Algorithm 2 Source Behavior

- The source starts to transmit at ACR := ICR, which is greater than or equal to its MCR;
- For every N_{rm} transmitted ATM data cells, the source sends a forward RM(CCR, MCR, ER) cell with CCR := ACR; MCR := MCR; ER := PCR;
- Upon the receipt a backward RM(CCR, MCR, ER) from the destination, the ACR at source is adjusted to: ACR := max{min{(ACR + AIR), ER, PCR}, MCR}. □

The destination end system simply returns every RM cell back towards the source upon receiving it.

The ATM Forum has not specified the ABR switch algorithm and has left its implementation to the vendors. In the following two sections, we present two ABR switch algorithms to achieve the MCRadd rate allocation policy.

4 A Simple Heuristic ABR Algorithm

The first implementation is based on the Intelligent Marking technique and does not require per-VC accounting [10, 11, 12].

4.1 The Intelligent Marking Technique

The key idea of the Intelligent Marking technique is to let each congested switch estimate max-min fair share rate for each VC bottlenecked at the switch with a small number of computations and without having the switch keeping track of each VC's state information (so called per-VC accounting). Fig. 3 illustrates the behavior of the Intelligent Marking technique. For each queue of a switch, four variables LOAD, MCCR (Mean CCR), UCR (Upper Cell Rate), and EBR (Estimated Bottleneck Rate) are defined. The value of LOAD corresponds to the aggregated cell rate entering the queue normalized with respect to link capacity and is measured by the switch over a period of time. The value of MCCR contains an estimated average cell rate of all VCs traversing this queue; the value of UCR contains an estimated upper limit on the cell rate of all VCs traversing this queue; and the value of EBR contains an estimated bottleneck rate at this queue. Furthermore, two parameters TLR and α are defined for each queue, where the value of TLR is the target load ratio, and $0 < \alpha < 1$.

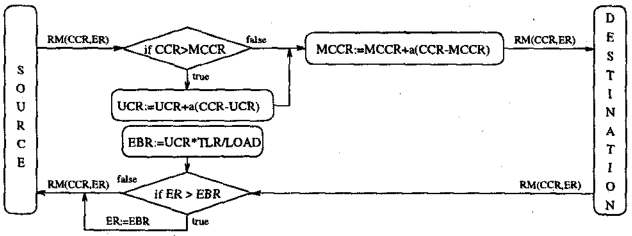

Figure 3: Switch behavior of Intelligent Marking protocol.

The Intelligent Marking algorithm is a heuristic algorithm. We will give an intuitive explanation on how it works. The RM cells from all VCs participate in exponential averaging for MCCR with MCCR := MCCR + α(CCR − MCCR) while only some VCs with greater than average rate (potentially VCs bottlenecked at this switch) participate in UCR averaging, which is used to estimate the bottleneck link rate. It has been shown in [11, 12] that this algorithm offers satisfactory performance in achieving max-min fair rate allocation for a variety of network configurations.

4.2 MCRadd Intelligent Marking

Since the Intelligent Marking technique allocates the max-min fair rate for each VC from network bandwidth when there is no MCR requirement [11, 12] and our MCRadd policy allocates each VC with MCR plus a max-min fair share from the *remaining* network capacity (subject to each session's PCR constraint), we can let the MCR-offsetted cell rate, CCR − MCR of each VC participate in Intelligent Marking and estimate the MCRadd-bottleneck link rate from the remaining network bandwidth.

Fig. 4 illustrates the switch behavior under the MCRadd Intelligent Marking technique. For each queue of a switch, four variables named LOAD, MFSR (Mean Fair Share Rate), UFSR (Upper Fair Share Rate), and EBR (Estimated Bottleneck Rate) are defined. The LOAD is the same as before. The value of MFSR contains an estimated MCR-offsetted average rate of all VCs traversing this queue; the UFSR contains an estimated MCR-offsetted upper rate; and the value of EBR contains an estimated MCRadd-bottleneck link rate. The parameters TLR and α are defined the same as before.

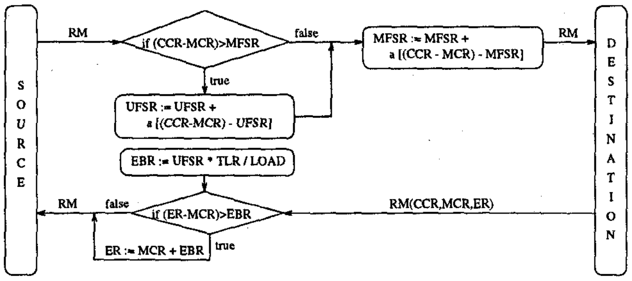

Figure 4: ABR switch behavior for the MCRadd policy.

4.3 Simulation Results for LANs

Here we present a simulation study demonstrating the effectiveness of our heuristic ABR algorithms to achieve the MCRadd policy. Table 2 lists the parameters used in our simulation. The distance from source/destination to the switch is 100 m and the link distance between ATM switches is 10 km. We assume that the propagation delay is 5 μs per km.

End System	PCR	PCR
	MCR	MCR
	ICR	MCR
	Nrm	32
	AIR	3.39 Mbps
Link	Speed	150 Mbps
Switch	Cell Switching Delay	4 μSec
	α	0.125

Table 2: Simulation parameters.

Peer-to-Peer Configuration

Fig. 6 shows the ACR at source for sessions s1, s2 and s3, respectively. The cell rates shown in the plot are normalized with respect to the link rate (150 Mbps) for easy comparison with those values obtained with our centralized algorithm under unit link capacity (Table 1). After the initial transient period, we see that the cell rate of each VC matches the rate listed in Table 1. We also show the inter-switch link utilization (Link 12) and queue size of congested switch (SW1) in Fig. 7. We find that the link is 100% utilized with small buffer requirements.

The Parking Lot Network Configuration

The parking lot configuration that we use is shown in Fig. 5 [6], where VC sessions $s1$ and $s2$ start from the first switch and go to the last switch. $s3$ and $s4$ start from SW2 and SW3, respectively, and terminate at the last switch.

Table 3 lists the MCR requirement and PCR constraint for each session and the rate assignment for each session under the centralized MCRadd rate allocation algorithm.

Session	MCR	PCR	MCRadd Rate Allocation
$s1$	0.15	0.20	0.20
$s2$	0.10	0.25	0.25
$s3$	0.10	0.50	0.30
$s4$	0.05	0.50	0.25

Table 3: MCR requirement, PCR constraint, and MCRadd rate allocation for each session in the parking-lot network configuration.

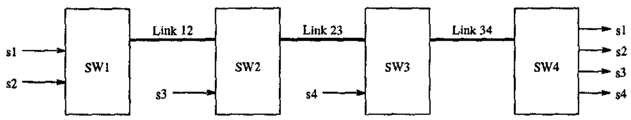

Figure 5: The parking-lot network configuration.

Fig. 8 shows the normalized cell rates of each VC under our ABR algorithm. We see that they match the rates listed in Table 3, which are obtained through the MCRadd centralized algorithm. Fig. 9 shows the link utilizations of Link34 and the output port buffer occupancy of SW3 for the same simulation run. Again, the network is 100% utilized with small buffer occupancy.

Our simulation results show that the rate allocation by our simple ABR algorithm achieves the MCRadd policy in a LAN environment. For a wide area network, a heuristic algorithm such as ours usually requires careful system parameter tuning to minimize oscillations. A more sophisticated ABR algorithm requiring per-VC accounting such as the one in the next section will be much more effective for a WAN. But in a LAN environment, where implementation cost is critical, our simple ABR algorithm here offers satisfactory performance with minimum implementation complexity.

5 A Second ABR Algorithm Using Per-VC Accounting

Our second ABR implementation for MCRadd fairness policy is based on the work in [3].

5.1 The ABR Algorithm

For each RM cell traversing this link, the switch records the CCR and MCR for each VC and performs the switch algorithm (Algorithm 4) at this link. Each link $\ell \in \mathcal{L}$ maintains a variable called advertised rate, μ_ℓ, which is used to estimate the MCRadd-bottleneck rate at this link.

The following are the link parameters and variables used in our switch algorithm.

C_ℓ**:** Capacity of link ℓ, $\ell \in \mathcal{L}$.

$\mathcal{G}_\ell$**:** Set of known sessions traversing link ℓ, $\ell \in \mathcal{L}$.

n_ℓ**:** Number of sessions in $\mathcal{G}_\ell$, $\ell \in \mathcal{L}$, i.e., $n_\ell = |\mathcal{G}_\ell|$.

r^i_ℓ**:** CCR value of session $i \in \mathcal{G}_\ell$ at link ℓ.

MCR^i**:** MCR requirement of session i.

b^i_ℓ**:** Bit used to mark session $i \in \mathcal{G}_\ell$ at link ℓ.

$$b^i_\ell = \begin{cases} 1 & \text{if session } i \in \mathcal{G}_\ell \text{ is marked at link } \ell; \\ 0 & \text{otherwise.} \end{cases}$$

$\mathcal{Y}_\ell$**:** Set of marked sessions at link ℓ, i.e. $\mathcal{Y}_\ell = \{i \mid i \in \mathcal{G}_\ell \text{ and } b^i_\ell = 1\}$.

$\mathcal{U}_\ell$**:** Set of unmarked sessions at link ℓ, i.e. $\mathcal{U}_\ell = \{i \mid i \in \mathcal{G}_\ell \text{ and } b^i_\ell = 0\}$ and $\mathcal{Y}_\ell \cup \mathcal{U}_\ell = \mathcal{G}_\ell$.

μ_ℓ**:** Advertised MCRadd-bottleneck link rate at link ℓ, calculated as follows:

Algorithm 3 μ_ℓ **Calculation**

$$\mu_\ell := \begin{cases} C_\ell & \text{if } n_\ell = 0; \\ C_\ell - \sum_{i \in \mathcal{G}_\ell} r^i_\ell \\ \quad + \max_{i \in \mathcal{G}_\ell}(r^i_\ell - \mathrm{MCR}^i) & \text{if } n_\ell = |\mathcal{Y}_\ell|; \\ \frac{(C_\ell - \sum_{i \in \mathcal{G}_\ell} \mathrm{MCR}^i) - \sum_{i \in \mathcal{Y}_\ell}(r^i_\ell - \mathrm{MCR}^i)}{|\mathcal{U}_\ell|} & \text{otherwise.} \end{cases}$$

□

Link initialization: $\mathcal{G}_\ell = \emptyset$; $n_\ell = 0$; $\mu_\ell = C_\ell$.

Algorithm 4 Switch Behavior

Upon the receipt of a forward RM(CCR, MCR, ER) cell from the source of session i {
    if RM cell signals session termination[2]{
        $\mathcal{G}_\ell := \mathcal{G}_\ell - \{i\}$;
        $n_\ell := n_\ell - 1$;
        table_update();
        }
    if RM cell signals session initiation {
        $\mathcal{G}_\ell := \mathcal{G}_\ell \cup \{i\}$;
        $n_\ell := n_\ell + 1$;
        $b^i_\ell := 0$; $r^i_\ell :=$ CCR; $\mathrm{MCR}^i :=$ MCR;
        table_update();
        }
    else /* i.e. RM cell belongs to a session already known at the link */ {
        $r^i_\ell :=$ CCR;
        if $((r^i_\ell - \mathrm{MCR}^i) \le \mu_\ell)$ then $b^i_\ell := 1$;
        table_update();
        }
    Forward RM(CCR, MCR, ER) towards its destination;
}

[2]This information is conveyed through some unspecified bits in the RM cell, which can be set either at the source or the UNI.

Upon the receipt of a backward RM(CCR, MCR, ER) cell from the destination of session i {
    ER := min{ER, $\mu_\ell + \text{MCR}^i$};
    Forward RM(CCR, MCR, ER) towards its source;
}

table_update()
{
    rate_calculation_1: use Algorithm 3 to calculate μ_ℓ^1;

    Unmark any session $i \in \mathcal{G}_\ell$ at link ℓ with $r_\ell^i - \text{MCR}^i > \mu_\ell^1$;

    rate_calculation_2: use Algorithm 3 to calculate μ_ℓ again;[3]
} □

Theorem 3 After the number of sessions in the network stabilizes, the rate allocation for each session by Algorithm 4 converges to the MCRadd fair rate allocation policy. □

The proof of Theorem 3 is given in [4]. Theorem 3 gives us a theoretical guarantee that this ABR algorithm will converge to the MCRadd policy under *any* network configuration and *any* set of link distances.

5.2 Simulation Results for WANs

The simulation parameters are the same as Table 2 except that we set AIR to PCR. This will make the ACR at source be set to ER upon receiving a returning RM cell (see Algorithm 2). For stability, we set the target link utilization to be 0.95. That is, we set $C_\ell = 0.95 \times 150$ Mbps $= 142.5$ Mbps at every link $\ell \in \mathcal{L}$ for ER calculation. The distance from source/destination to the switch is 1 km and the link distance between ATM switches is 1000 km.

The Peer-to-Peer Network Configuration

Fig. 10 shows the ACR at source for sessions s1, s2 and s3, respectively. The cell rates shown in the plot are normalized with respect to the capacity C_ℓ (142.5 Mbps) for easy comparison with those values obtained with our centralized algorithm under unit link capacity (Table 1). After initial iterations, we see that the cell rate of each session converges to the final rate listed in Table 1.

The Parking Lot Network Configuration

Fig. 11 shows the normalized cell rates of each session under our ABR implementation. We see that they converge to the rates listed in Table 3 after initial iterations.

6 Concluding Remarks

We have defined a network bandwidth assignment policy to support MCR/PCR requirement for ABR service. Two ER-based ABR algorithms have been developed in the context of the ATM Forum ABR traffic management framework to achieve the MCRadd fair rate allocation policy. The first ABR implementation does not require per-VC accounting and has $O(1)$ storage and computational complexity. Its effectiveness to achieve the MCRadd policy in LAN environment has been demonstrated by simulation results. The second ABR implementation requires per-VC accounting and is proven to converge to MCRadd rate allocation policy under any network topology and any set of link distance. It has $O(N)$ storage and computational complexity, where N is the number of VCs in the network, and would be suitable for implementation in an ATM WAN switch.

[3] Both μ_ℓ^1 and μ_ℓ follow the same rate calculation in Algorithm 3.

References

[1] ATM Forum Technical Committee, "Traffic Management Specification - Version 4.0," *ATM Forum/95-0013R13,* February 1996.

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

[3] A. Charny, D. Clark, and R. Jain, "Congestion Control with Explicit Rate Indication," *Proc. IEEE ICC'95,* pp. 1954-1963.

[4] Y. T. Hou, H. Tzeng, and S. S. Panwar, "The MCRadd Rate Allocation Policy for ABR Service," *Technical Report CATT 97-104,* Center for Advanced Technology in Telecommunications, Polytechnic University, Brooklyn, NY, Jan. 20, 1997.

[5] D. Hughes, "Fair Share in the Context of MCR," *ATM Forum Contribution 94-0977,* October 1994.

[6] R. Jain, "Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey," *Computer Networks and ISDN Systems,* February 1995.

[7] R. Jain, *et al.*, "ERICA Switch Algorithm: A Complete Description," *ATM Forum Contribution, 96-1172,* August 1996.

[8] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, "Dynamics of an Explicit Rate Allocation Algorithm for Available Bit Rate (ABR) Service in ATM Networks," *Proc. 6th IFIP Int. Conf. High Performance Networking (HPN '95),* Sept. 1995, pp.143-154.

[9] L. Roberts, "Enhanced PRCA (Proportional Rate Control Algorithm," *ATM Forum Contribution 94-0735R1,* August 1994.

[10] K.-Y. Siu and H.-Y. Tzeng, "Intelligent Congestion Control for ABR Service in ATM Networks," *ACM SIGCOMM Computer Communication Review,* 24(5):81–106, October 1994.

[11] K.-Y. Siu and H.-Y. Tzeng, "Limits of Performance in Rate-based Control Schemes," *ATM Forum Contribution 94-1077,* November 1994.

[12] H.-Y. Tzeng and K.-Y. Siu, "Comparison of Performance Among Existing Rate Control Schemes," *ATM Forum Contribution 94-1078,* November 1994.

[13] N. Yin, "Max-Min Fairness vs. MCR Guarantee on Bandwidth Allocation for ABR," *Proc. IEEE ATM'96 Workshop,* San Francisco, CA, August 25-27, 1996.

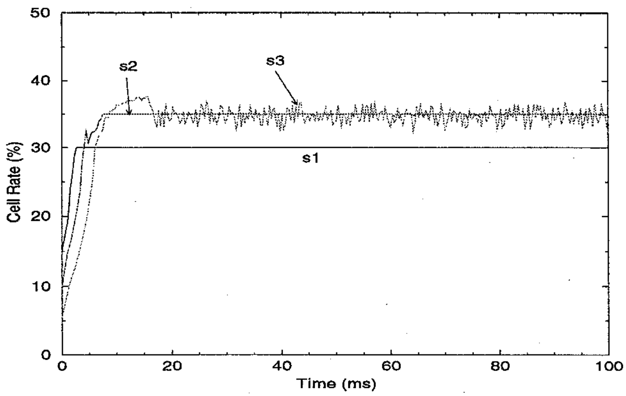

Figure 6: The cell rates of all connections for the MCRadd policy in the peer-to-peer network configuration under the first ABR algorithm.

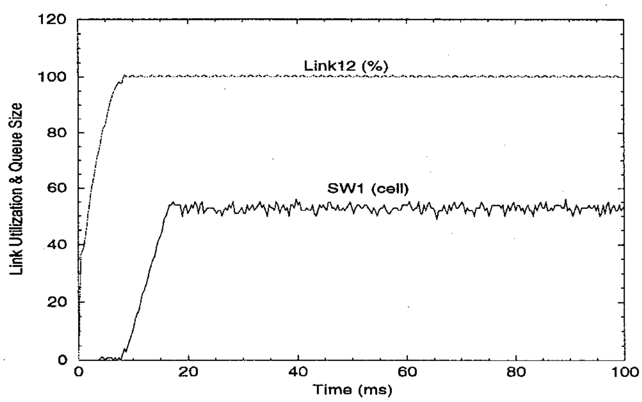

Figure 7: The link utilization and the queue size of the congested switch for the MCRadd policy in the peer-to-peer network configuration under the first ABR algorithm.

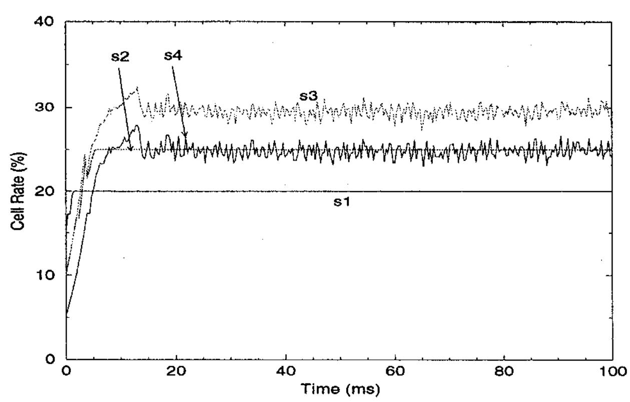

Figure 8: The cell rates of all connections for the MCRadd policy in the parking-lot network configuration under the first ABR algorithm.

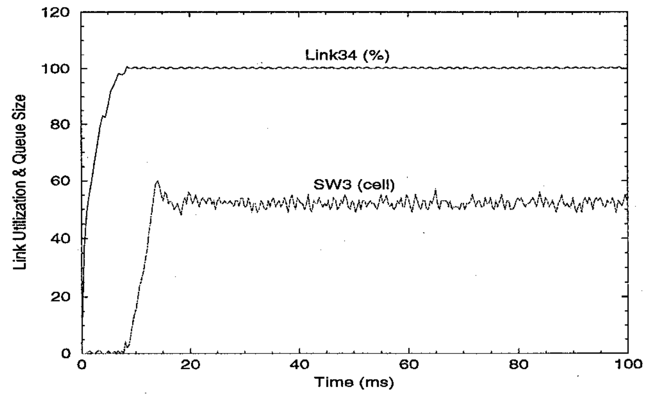

Figure 9: The link utilization and the queue size of the congested switch (SW3) for the MCRadd policy in the parking-lot network configuration under the first ABR algorithm.

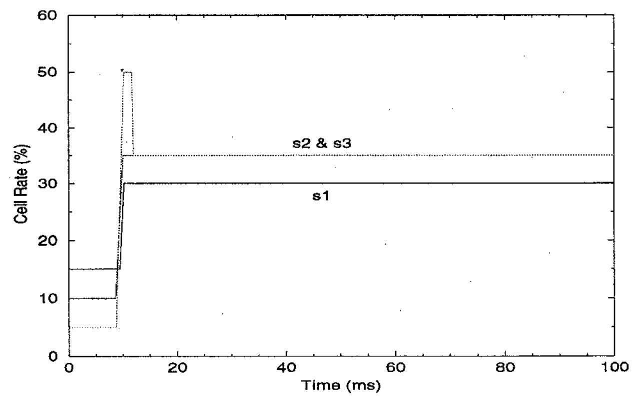

Figure 10: The cell rates of all connections for the MCRadd policy in the peer-to-peer network configuration under the second ABR algorithm.

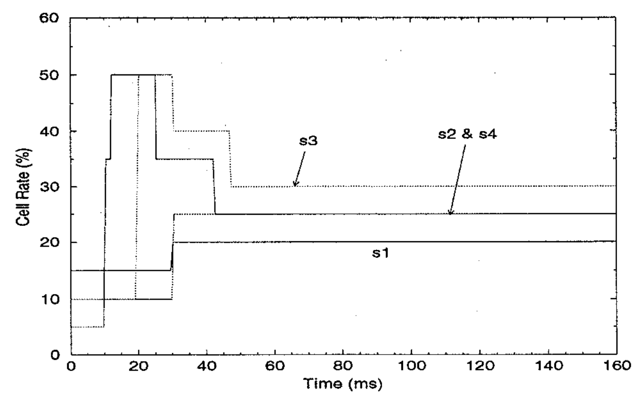

Figure 11: The cell rates of all connections for the MCRadd policy in the parking lot network configuration under the second ABR algorithm.