Submitted to IEICE Trans. on Comm. Special Issue on ATM Traffic Control and Performance Evaluation.

ATM ABR Traffic Control with a Generic Weight-Based Bandwidth Sharing Policy: Theory and a Simple Implementation *

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

Abstract

The ATM Forum has suggested the use of the classical max-min rate allocation policy for the available bit rate (ABR) service class. However, there are several drawbacks in adopting the max-min policy. In particular, the max-min policy is not able to support the minimum cell rate (MCR) guarantee and the peak cell rate (PCR) constraint for each connection, which are part of the ABR service specifications. Furthermore, it is not flexible enough for network providers wishing to establish pricing strategies based on the rate allocation.

To support MCR/PCR for all connections in any network, we present a generic weight-based max-min (WMM) policy, which generalizes the classical max-min policy and offers flexible pricing options. A centralized algorithm is presented to compute network-wide bandwidth allocation to achieve this policy. Furthermore, a simple switch algorithm using ABR flow control protocol is developed with the aim of achieving the WMM rate allocation policy in a distributed networking environment. The effectiveness of our distributed algorithm is demonstrated by simulation results based on the benchmark network configurations suggested by the ATM Forum.

Key Words: Max-Min Policy, Minimum Rate Requirement, Peak Rate Constraint, Centralized and Distributed Algorithms, Heuristics, Congestion/Flow Control, Packet-Switched Networks, ABR Traffic Control, ATM Networks.

---

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

[†] Fujitsu Laboratories of America, Inc., 3350 Scott Blvd., Bldg. #34, Santa Clara, CA 95054-3104, Tel. (408) 567-4529, Fax (408) 567-4558, Email: thou@fla.fujitsu.com.

[‡] Bell Laboratories, Lucent Technologies, Room 4C-532, 101 Crawfords Corner Road, Holmdel, NJ 07733. Tel. (732) 949-5969, Fax (732) 834-5354, Email: htzeng@research.bell-labs.com.

[§] Please direct all correspondence to Prof. Shivendra S. Panwar, Dept. of Electrical Engineering, Polytechnic University, Six Metrotech Center, Brooklyn, NY 11201. Tel. (718) 260-3740, Fax (718) 260-3074, Email: panwar@kanchi.poly.edu.

[¶] Bell Laboratories, Lucent Technologies, Room 4D-533, 101 Crawfords Corner Road, Holmdel, NJ 07733. Tel. (732) 949-2014, Fax (732) 949-7575, Email: vijay@research.bell-labs.com.

1 Introduction

The available bit rate (ABR) service class defined by the ATM Forum supports applications that allow the ATM source end system to adjust the information transfer rate based on the bandwidth availability in the network [1]. On the establishment of an ABR connection, the user shall specify to the network both minimum and maximum rate requirements, designated as the minimum cell rate (MCR) and the peak cell rate (PCR), respectively. 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 the choice of a network bandwidth sharing policy among competing user connections. The ATM Forum has suggested the use of the classical max-min policy to share network bandwidth among ABR connections [2]. Informally, the max-min policy attempts to maximize the smallest rate among all connections; given the best smallest rate allocation, the second smallest rate allocation is maximized, etc.

There are a few drawbacks associated with using the classical max-min policy for ABR service. First of all, the max-min policy does not support the MCR/PCR constraints of each connection. Secondly, the max-min policy treats each session with equal priority and thus is not flexible enough for network providers wishing to set up a flexible pricing policy consistent with the bandwidth allocation policy.

This paper presents a generic weight-based max-min (WMM) policy, which generalizes the classical max-min policy. Our policy supports MCR/PCR constraints for each session and offers meaningful and flexible pricing strategies for network providers. This policy is a generalization of several policies described in [7, 8, 11, 23]. In particular, we assign a weight to each connection. Such weight assignment is decoupled (i.e. independent) from the MCR requirement and PCR constraint of the connection. The bandwidth is first allocated to each connection based its MCR requirement. Then the excessive bandwidth is allocated proportionally to the weight of each connection while not exceeding its PCR constraint and link capacity constraint.

We first present a centralized algorithm to achieve our WMM policy. This centralized algorithm requires global information, which is usually difficult to maintain in a distributed environment. We are interested in the design of a distributed networking protocol to achieve the WMM policy through distributed computation in the absence of global knowledge about the network, and without the synchronization of different network components.

We employ the ATM Forum ABR flow control protocol to design our distributed algorithm. In our previous work [7, 8], we presented a few simple algorithms to support several special cases of the WMM policy, e.g. uniform weight for each connection [8] and MCR-based weight assignment for each connection [7]. This paper generalizes our previous work by presenting a distributed algorithm for the WMM policy under any weight assignment for each connection. Our algorithm is based on the *Intelligent Marking* technique by Siu and Tzeng [20], which achieves the classical max-min policy. We extend this technique to design a distributed algorithm for our WMM policy.

Since our distributed algorithm for the WMM policy is a heuristic algorithm, we perform simulations on a few benchmark network configurations suggested by the ATM Forum Traffic Management Group to demonstrate its effectiveness.

The remainder of this paper is organized as follows. In Section 2, we define our generic weight-based max-min (WMM) policy. We also present a centralized algorithm for the WMM policy. In Section 3, we design a simple distributed algorithm using the ABR flow control protocol to achieve the WMM policy. In Section 4, we present simulation results of our distributed algorithm for various network configurations. Section 5 concludes this paper and points out future research directions.

2 A Generic Weight-Based Rate Allocation Policy

This section presents the centralized theory of the generic weight-based max-min (WMM) network bandwidth sharing policy.

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 (maximum allowable bandwidth) of link ℓ. A link ℓ is *saturated* or *fully utilized* if $F_\ell = C_\ell$.

Let MCR_s and PCR_s be the minimum rate requirement and the peak rate constraint for each session $s \in \mathcal{S}$.

Definition 1 A rate vector $r = (\cdots, r_s, \cdots)$ is *ABR-feasible* if the following two constraints are satisfied:

$$\begin{aligned} \text{MCR}_s \le r_s \le \text{PCR}_s \quad & \text{for all } s \in \mathcal{S}; \\ F_\ell \le C_\ell \quad & \text{for all } \ell \in \mathcal{L}. \end{aligned}$$

□

---

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

For the sake of feasibility, we make the following assumption.

Assumption 1 The sum of all sessions' MCR requirements traversing any link does not exceed the 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 can be enforced by admission control at call setup time to determine whether or not to accept a new connection.

We assign each session $s \in \mathcal{S}$ with a weight (or priority) w_s.[2] The WMM policy first allocates to each session its MCR. Then from the remaining network capacity, it allocates additional bandwidth for each session using a proportional version of the max-min policy based on each session's weight while satisfying the PCR constraint and the link capacity constraint. The final bandwidth allocated to each session is its MCR plus an additional "weighted" max-min share. Formally, this policy is defined as follows.

Definition 2 A rate vector r is *weight-based max-min (WMM)* 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 $\frac{r_s - \text{MCR}_s}{w_s} \ge \frac{r_t - \text{MCR}_t}{w_t}$ and $r_t > \hat{r}_t$. □

We define a new notion of WMM-bottleneck link in the following definition.

Definition 3 Given an ABR-feasible rate vector r, a link $\ell \in \mathcal{L}$ is a *WMM-bottleneck link* with respect to r for a session s traversing ℓ if $F_\ell = C_\ell$ and $\frac{r_s - \text{MCR}_s}{w_s} \ge \frac{r_t - \text{MCR}_t}{w_t}$ for all sessions t traversing link ℓ. □

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

Please refer to the Appendix for the proof of Theorem 1.

The following uniqueness property has been shown in [9].

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

The following WMM centralized algorithm computes the rate allocation for each session $s \in \mathcal{S}$ in any network $\mathcal{N}$ such that the WMM policy is satisfied. Informally, the algorithm works as following:

---

[2] We assume a positive weight assignment for each connection.

1. Start the rate allocation of each session with its MCR.
2. Increase the rate of each session with an increment proportional to its weight until 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 capacity 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.

Formally, the WMM centralized algorithm is stated as following.

Algorithm 1 A Centralized Algorithm for the WMM Policy

Initial conditions:

$k = 1$, $\mathcal{S}^1 = \mathcal{S}$, $\mathcal{L}^1 = \mathcal{L}$,
$r_s^0 = \mathrm{MCR}_s$, for every $s \in \mathcal{S}$,
$F_\ell^0 = \sum_{\text{all } s \in \mathcal{S} \text{ traversing } \ell} \mathrm{MCR}_s$, for every $\ell \in \mathcal{L}$.

1. $m^k := \min\{\min_{\ell \in \mathcal{L}^k} \frac{(C_\ell - F_\ell^{k-1})}{\sum_{s \in \mathcal{S}^k \text{ traversing } \ell} w_s}, \ \min_{s \in \mathcal{S}^k} \frac{(\mathrm{PCR}_s - r_s^{k-1})}{w_s}\}.$
2. $a_s^k := m^k \times w_s$, for every $s \in \mathcal{S}^k$.
3. $r_s^k := \begin{cases} r_s^{k-1} + a_s^k & \text{if } s \in \mathcal{S}^k; \\ r_s^{k-1} & \text{otherwise.} \end{cases}$
4. $F_\ell^k := \sum_{\text{all } s \in \mathcal{S} \text{ traversing } \ell} r_s^k$, for every $\ell \in \mathcal{L}^k$.
5. $\mathcal{L}^{k+1} := \{\ell \mid C_\ell - F_\ell^k > 0, \ell \in \mathcal{L}^k\}$.
6. $\mathcal{S}^{k+1} := \{s \mid s$ does not traverse any link $\ell \in (\mathcal{L} - \mathcal{L}^{k+1})$ and $r_s^k \neq \mathrm{PCR}_s\}$.
7. $k := k + 1$.
8. If $\mathcal{S}^k$ is empty, then $r^{k-1} = (\cdots, r_s^{k-1}, \cdots)$ is the rate vector satisfying the WMM policy and this algorithm terminates; otherwise, go back to Step 1. □

The correctness proof that Algorithm 1 achieves the WMM rate allocation is given in the Appendix.

The following examples illustrate how Algorithm 1 allocates network bandwidth such that the WMM policy is satisfied.

Example 1 Peer-to-Peer Configuration

In this network configuration (Fig. 1), the output port link of SW1 (Link 12) is the only potential bottleneck link for all sessions.

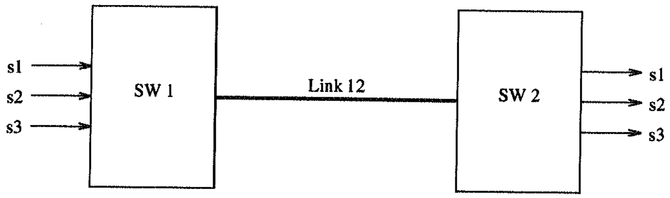


Figure 1: The peer-to-peer network configuration.

Assume that all links are of unit capacity. The MCR requirement and the PCR constraint for each session are listed in Table 1.

Session	MCR Requirement	PCR Constraint	WMM Rate Allocation		
			$w = 1$	$w = \text{MCR}$	$w = \text{PCR} - \text{MCR}$
$s1$	0.15	1.00	0.40	0.525	0.547
$s2$	0.10	0.30	0.30	0.300	0.193
$s3$	0.05	0.50	0.30	0.175	0.260

Table 1: MCR requirement, PCR constraint, and WMM rate allocation for each session under three different weight functions in the peer-to-peer network configuration.

We show the rate allocation for each session under the following three cases of weight assignment in Table 1.

- *Case 1:* Equal weight for all sessions, $w_1 = w_2 = w_3$.

  In this case, the rate allocation for each session is its MCR plus a max-min share of the remaining network capacity.

- *Case 2:* Each session's weight is proportional to its MCR, i.e. $w_i \propto \text{MCR}_i$, $i = 1, 2, 3$.
  In this case, the WMM policy becomes the MCR-weighted max-min policy. Such a policy may be used when the charge or price of a session is proportional to its MCR.

- *Case 3:* Each session's weight is proportional to the difference between its PCR and MCR, i.e. $w_i \propto (\text{PCR}_i - \text{MCR}_i)$, $i = 1, 2, 3$.
  The objective of this policy is to allocate each session the same fraction of the difference between its PCR and MCR, everything else being the same.

As an illustration, we show the iterative steps of using Algorithm 1 for Case 2 in Table 2.

Iterations	Session(MCR, PCR)			Remaining Capacity
	$s1(0.15, 1.00)$	$s2(0.10, 0.30)$	$s3(0.05, 0.50)$	Link 12
initialization	0.15	0.10	0.05	0.70
1st	0.45	$\boxed{0.30}$	0.15	0.10
2nd	$\boxed{0.525}$		$\boxed{0.175}$	0

Table 2: Iterations of using the WMM centralized algorithm to calculate rate allocation for each session in the peer-to-peer network configuration. We assume each session's weight is proportional to its MCR, i.e. $w_i \propto \text{MCR}_i$, $i = 1, 2, 3$ (Case 2).

In general, the weight assignment of a session can be arbitrary and we shall obtain a unique corresponding WMM rate allocation (as shown in Theorem 2). □

Remark 1 It should be clear by now that our WMM policy provides a meaningful and flexible pricing strategy for network providers. In particular, each connection may be charged a basic rate corresponding to the guaranteed bandwidth (i.e. MCR). Beyond this rate, each connection may pay an additional tariff for a weight (or priority) to share any additional (unguaranteed or available) network capacity. □

The following example illustrates WMM rate allocation in a multi-node network.

Example 2 The Three-Node Network Configuration

In this network configuration (Fig. 2), the output port links of SW1 (Link 12) and SW2 (Link 23) are the potential bottleneck links for all sessions.

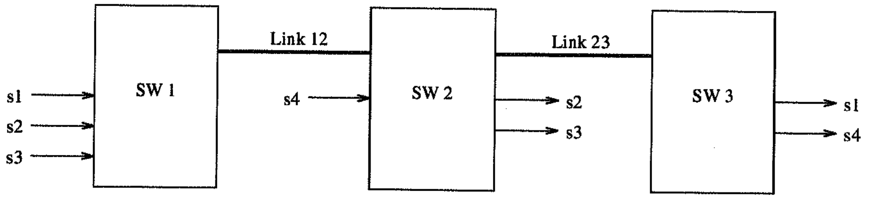


Figure 2: The three-node network configuration.

The MCR requirement and PCR constraint for each session are listed in Table 3, as well as the WMM rate allocation for each session under three cases of weight assignments.

As an illustration, in Table 4, we list the iterative steps of using Algorithm 1 for the case where each session's weight is proportional to its MCR. □

The centralized theory on the WMM policy requires global information, which is usually difficult to maintain in a distributed environment. We are interested in the design of a distributed networking protocol to achieve the WMM policy through distributed computation in the absence of global

Session	MCR Requirement	PCR Constraint	WMM Rate Allocation		
			$w = 1$	$w = \mathrm{MCR}$	$w = \mathrm{PCR} - \mathrm{MCR}$
$s1$	0.05	0.75	0.25	0.15	0.304
$s2$	0.15	0.90	0.35	0.45	0.423
$s3$	0.20	0.40	0.40	0.40	0.273
$s4$	0.10	1.00	0.75	0.85	0.696

Table 3: MCR requirement, PCR constraint, and WMM rate allocation for each session under various weight functions in the three-node network configuration.

Iterations	Session(MCR, PCR)				Remaining Capacity	
	$s1$ (0.05, 0.75)	$s2$ (0.15, 0.90)	$s3$ (0.20, 0.40)	$s4$ (0.10, 1.00)	Link 12	Link 23
initialization	0.05	0.15	0.20	0.10	0.60	0.85
1st	0.10	0.30	$\boxed{0.40}$	0.20	0.20	0.70
2nd	$\boxed{0.15}$	$\boxed{0.45}$		0.30	0	0.55
3rd				$\boxed{0.85}$		0

Table 4: Iterations of using the WMM centralized algorithm to calculate rate allocation for each session in the three-node network configuration. Case 2: each session's weight is proportional to its MCR, $w_i \propto \mathrm{MCR}_i$, $i = 1, 2, 3, 4$.

knowledge about the network and without the synchronization of different network components. This will be our focus in the next section.

3 A Simple Distributed Protocol for the WMM Policy

Previous Work

There have been extensive prior efforts on the design of distributed algorithms to achieve the classical max-min policy [5, 12, 4, 16, 17]. In essence, all these schemes maintain some link controls at the switch level and convey some information about these controls to the source by means of feedback. Upon the receipt of the feedback signal, the source adjusts its estimate of the allowed transmission rate according to some rule. These algorithms essentially differ in the particular choices of link controls and the type of feedback provided to the sources by the network.

Recent research efforts on the ABR service at the ATM Forum has brought many renewed efforts to the design of distributed algorithms to achieve the classical max-min policy. These algorithms either used heuristics [14, 15, 18, 19, 22] or a theoretical approach [3] and had different performance behaviors and implementation complexities. In particular, the *Intelligent Marking* technique by Siu and Tzeng, originally proposed in [19], and further refined in [20, 21], offers satisfactory performance in achieving the classical max-min policy with minimal implementation complexity. It has been adopted by the ATM Forum Traffic Management Group as one of the standard options for ABR switch algorithms. This paper extends this technique to design a distributed algorithm to achieve our WMM policy with MCR/PCR support and a weight assignment for each connection.

Since our distributed protocol uses the ATM ABR flow control mechanism, we briefly summarize the relevant terminology for ABR before we present our algorithm.

3.1 ABR Flow Control Mechanism

The key idea in the ABR flow control mechanism is to use cooperation between the sources and the network. In particular, such cooperation includes the following components:

1. Information exchange:
    - The source informs the network about its rate information;
    - The network informs the source about network bandwidth.
2. Source rate adaptation:
    - The source adjusts its rate based on the feedback information from the network.

A generic ABR flow control mechanism for a virtual connection is shown in Fig. 3. To achieve information exchange, Resource Management (RM) cells are inserted among data cells to convey information between the source and the network. The source sets the fields in the forward RM cells to inform the network about the source's rate information (e.g. MCR, PCR, CCR). For each traversing RM cell at a switch, the switch performs some calculations based on the information carried in this RM cell and at the switch. We let the network (switches) set the fields in the backward RM cells to inform the source. To achieve source rate adaptation, the source adjusts its transmission rate upon receiving a backward RM cell.

3.2 A Simple Distributed Implementation

We first specify the behaviors of each connection's source and destination in our algorithm, which conforms to the framework of the ABR specifications [1].

Algorithm 2 End System Behavior

Source Behavior[3]

The source starts to transmit at ACR := ICR, which is greater than or equal to its MCR;

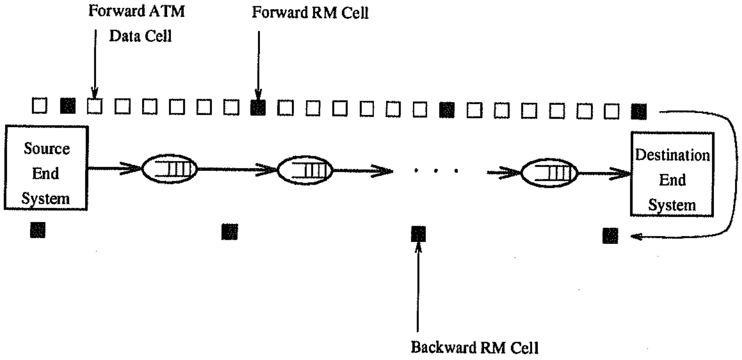


Figure 3: Closed-loop rate-based flow control for an ABR virtual connection.

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 of a backward RM(CCR, MCR, ER) cell from the destination, the ACR at the source is adjusted to:

$$\text{ACR} := \max\{\min\{(\text{ACR} + \text{AIR}),\ \text{ER}\},\ \ \text{MCR}\}.$$

Destination Behavior

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

Our switch algorithm for the WMM policy is based on the Intelligent Marking technique for the classical max-min policy, originally proposed by Siu & Tzeng in [19] and further refined in [20, 21]. The key idea of this technique is to employ a small number of variables and a small number of computations at each output port of a switch to estimate the max-min bottleneck link rate. Using a feedback mechanism, the ER field of a returning RM cell is set to the minimum of all the estimated bottleneck link rates on all its traversing links to achieve max-min fair share. The details of the Intelligent Marking technique are given in [20].

Since the WMM policy first allocates each session with its MCR, and then allocates the remaining network bandwidth to each session using the w-weighted max-min policy, this motivates us to let the CCR and ER fields of a traversing RM cell first offsetted by its MCR, and then normalized with respect to its weight w (i.e. $\frac{\text{CCR}-\text{MCR}}{w}$, $\frac{\text{ER}-\text{MCR}}{w}$) in order to participate in the Intelligent

---

[3] We use a simplified version of source and destination behavior, which does not include the use-it-or-lose-it option [1].

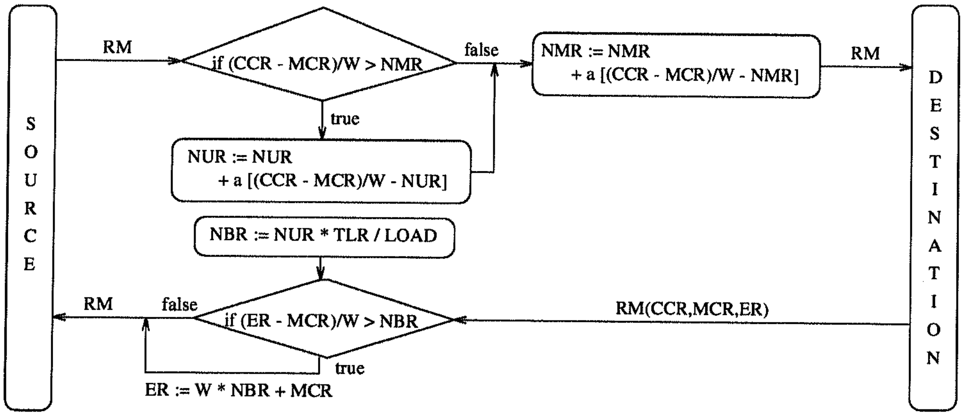


Figure 4: Switch behavior for the WMM policy.

Marking algorithm.

Fig. 4 illustrates our switch algorithm for the WMM policy. Four variables named LOAD, NMR (Normalized Mean Rate), NUR (Normalized Upper Rate) and NBR (Normalized Bottleneck Rate) are defined at each output port of an ATM switch. The value of LOAD corresponds to the aggregated cell rate entering the output queue normalized with respect to the link capacity. It is measured at the switch output port over a period of time. The value of NMR contains an exponential averaging of $(\text{CCR} - \text{MCR})/w$ for all VCs traversing this link; the value of NUR contains an exponential averaging of $(\text{CCR} - \text{MCR})/w$ only for VCs with $(\text{CCR} - \text{MCR})/w > \text{NMR}$; and NBR contains an estimated normalized WMM bottleneck link rate. Here, NMR, NUR and NBR are all dimensionless. TLR is the targeted load ratio and $0 < \alpha < 1$.

Algorithm 3 Switch Behavior

```
Upon the receipt of RM(CCR, MCR, ER) from the source of a VC
    if (CCR − MCR)/w > NMR, then
        NUR := NUR + $\alpha$ [(CCR − MCR)/w - NUR];
    NMR := NMR + $\alpha$ [(CCR − MCR)/w - NMR];
    Forward RM(CCR, MCR, ER) to its destination;

Upon the receipt of RM(CCR, MCR, ER) from the destination of a VC
    NBR := NUR * TLR / LOAD;
    if (QS > QT),[4] then
        NBR := (QT / QS) * NBR;
    if (ER − MCR)/w > NBR, then
        ER := w × NBR + MCR;
        Forward RM(CCR, MCR, ER) to its source;
    else
        Forward RM(CCR, MCR, ER) to its source.
```

□

By the operations of Algorithms 2 and 3, we have the following fact for the ACR parameter at the source and the CCR field in the RM cell.

Fact 1 For every connection $s \in \mathcal{S}$, the ACR at the source and the CCR field in the RM cell are ABR-feasible, i.e. $\text{MCR}^s \leq \text{ACR}^s \leq \text{PCR}^s$ and $\text{MCR}^s \leq \text{CCR}^s \leq \text{PCR}^s$. □

4 Simulation Results

In this section, we implement our switch algorithm on our network simulator [6] and perform a simulation study to demonstrate the effectiveness of our distributed algorithm in achieving the WMM rate allocation policy.

The network configurations that we use are the peer-to-peer and the three-node network configurations shown in Figs. 1 and 2, respectively, and the *parking lot* network configuration (Fig. 5). These network configurations have been used as standard configurations to test ABR flow control protocols by the ATM Forum Traffic Management Group.

The ATM switches in all the simulations are assumed to have output port buffers with a speedup equal to the number of their ports. Each output port buffer of a switch employs the simple FIFO queuing discipline and is shared by all VCs going through that port. At each output port of a switch, we implement our switch algorithm.

Table 5 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 (corresponding to a local area network) and 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
	Load/Utilization Measurement Interval	500 μSec
	Queue Threshold for ER Adjustment	50 cells
	Output Buffer Size	2000 cells

Table 5: Simulation parameters.

---

[4] This step is a finer adjustment of NBR calculation based on buffer occupancy information and is not shown in Fig. 4 due to space limitation. QS is the Queue Size of the output link and QT is a predefined Queue Threshold.

For each network configuration, we run simulations under the following three cases of weight functions:

- *Case 1:* $w_s = 1$, $s \in \mathcal{S}$, which corresponds to equal weight for all sessions.
- *Case 2:* $w_s = \mathrm{MCR}_s$, $s \in \mathcal{S}$, which corresponds to the scenario that each session's weight is proportional to its MCR.
- *Case 3:* $w_s = \mathrm{PCR}_s - \mathrm{MCR}_s$, $s \in \mathcal{S}$, which corresponds to the case where each session's weight is proportional to the difference between its PCR and MCR.

The Peer-to-Peer Network Configuration

In this network configuration (Fig. 1), the output port link of SW1 (Link 12) is the only potential bottleneck link for all sessions. Assume that all links are of unit capacity. The MCR requirement, the PCR constraint, and the WMM rate allocation for each session are listed in Table 1.

The upper plot in Fig. 6 shows the ACR at source for sessions s1, s2 and s3, respectively, for the case where each session's weight is 1. 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 rates of VCs match fairly well with the rates listed in Table 1. The ripples in the ACR curves for s_1 and s_3 are due to the dynamic nature of the queue size during a simulation run, which is used to adjust ER calculation (see Algorithm 3). The ACR curve for s_2 is fairly flat because of the PCR constraint for s_2 (see source behavior in Algorithm 2). To study the network utilization of our algorithm, we also show the inter-switch link utilization (Link 12) and queue size of congested switch (SW1) in the lower plot of Fig. 6. We find that the link is 100% utilized with reasonably small buffer requirements.

Simulation results for the the cases where each session's weight is $w_s = \mathrm{MCR}_s$, $s \in \mathcal{S}$, and $w_s = \mathrm{PCR}_s - \mathrm{MCR}_s$, $s \in \mathcal{S}$, are shown in Figs. 7 and 8, respectively. Again, the cell rates of VCs match fairly well with the rates listed in Table 1 with 100% link utilization and small buffer occupancy.

The Three-Node Network Configuration

In this network configuration (Fig. 2), the output port links of SW1 (Link 12) and SW2 (Link 23) are potential WMM-bottleneck links for sessions. The MCR requirement, the PCR constraint, and the WMM rate allocation for each session are listed in Table 3.

Simulation results for the cell rate of each session, the bottleneck link utilization and buffer occupancy are shown in Figs. 9, 10 and 11 under three different weight mapping functions. In all cases, we see that after the initial transient period, the cell rate of each session matches with the rate listed in Table 3. Also, the bottleneck links are 100% utilized with reasonably small buffer occupancies.

The Parking Lot Network Configuration

This configuration and its name is derived from theater parking lots, which consists of several parking areas connected via a single exit path [13]. The specific parking lot configuration that we use is shown in Fig. 5 where sessions $s1$ and $s2$ start from the first switch and go to the last switch. Sessions $s3$ and $s4$ start from SW2 and SW3, respectively, and terminate at the last switch.

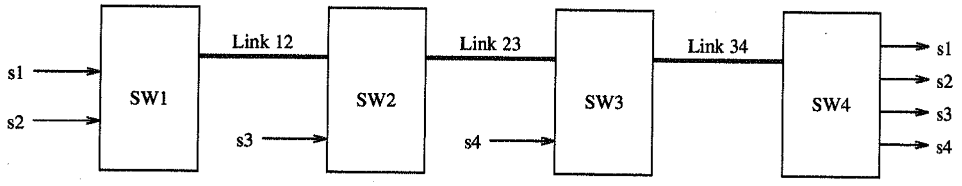


Figure 5: The parking lot network configuration.

Table 6 lists the MCR requirement and the PCR constraint for each session, and the rate assignment for each session under three weight assignments using the centralized WMM algorithm.

Session	MCR Requirement	PCR Constraint	WMM Rate Allocation		
			$w = 1$	$w = \text{MCR}$	$w = \text{PCR} - \text{MCR}$
$s1$	0.15	0.35	0.316	0.35	0.2543
$s2$	0.10	0.20	0.200	0.20	0.1522
$s3$	0.10	0.50	0.267	0.30	0.3087
$s4$	0.05	0.50	0.217	0.15	0.2848

Table 6: MCR requirement, PCR constraint, and WMM rate allocation for each session under three weighting functions in the parking-lot network configuration.

Simulation results for the cell rate of each session, the bottleneck link utilization and buffer occupancy are shown in Figs. 12, 13, and 14, respectively, for three different weight mapping functions. We see that the cell rates match fairly well with the rates listed in Table 6, which are obtained through the centralized WMM algorithm. Also the link utilizations of Link34 is 100% utilized with low buffer occupancy in the output port of SW3.

In summary, based on the simulation results in this section, we have demonstrated that our simple distributed algorithm achieves the WMM policy with minimal implementation complexity.

5 Concluding Remarks

We presented a generic weight-based max-min policy to allocate network bandwidth among connections. Our policy generalizes the classical max-min policy by supporting the MCR and PCR constraints for each connection and associating each connection with a weight. This policy offers meaningful and flexible pricing options for network providers wishing to set up a usage-based rate allocation and pricing policy. A centralized algorithm for the WMM policy was also presented.

We developed a simple heuristic algorithm using the ATM Forum ABR flow control protocol to achieve the WMM policy. Our algorithm uses simple heuristics based on the Intelligent Marking technique and does not require the switch to maintain a table and keep track of the state information of each traversing VC (so-called per-VC accounting). It has $O(1)$ storage requirements and computational complexity. Simulation results show that the rate allocation by our algorithm matches closely to the WMM policy in a LAN environment. For a wide area network (WAN), our simple heuristic algorithm requires careful system parameter tuning to minimize oscillation. Thus, a more sophisticated algorithm using per-VC accounting will be much more effective for a WAN [10]. But in a LAN environment, where implementation cost may well be the most important criterion in the choice of a switch algorithm, our algorithm offers satisfactory performance with minimal implementation complexity.

Our future work will focus on other issues of our distributed protocol for the WMM policy. Such issues include system transient behavior and network buffer requirements, all of which should be carefully investigated before a distributed algorithm is deployed in ATM networks.

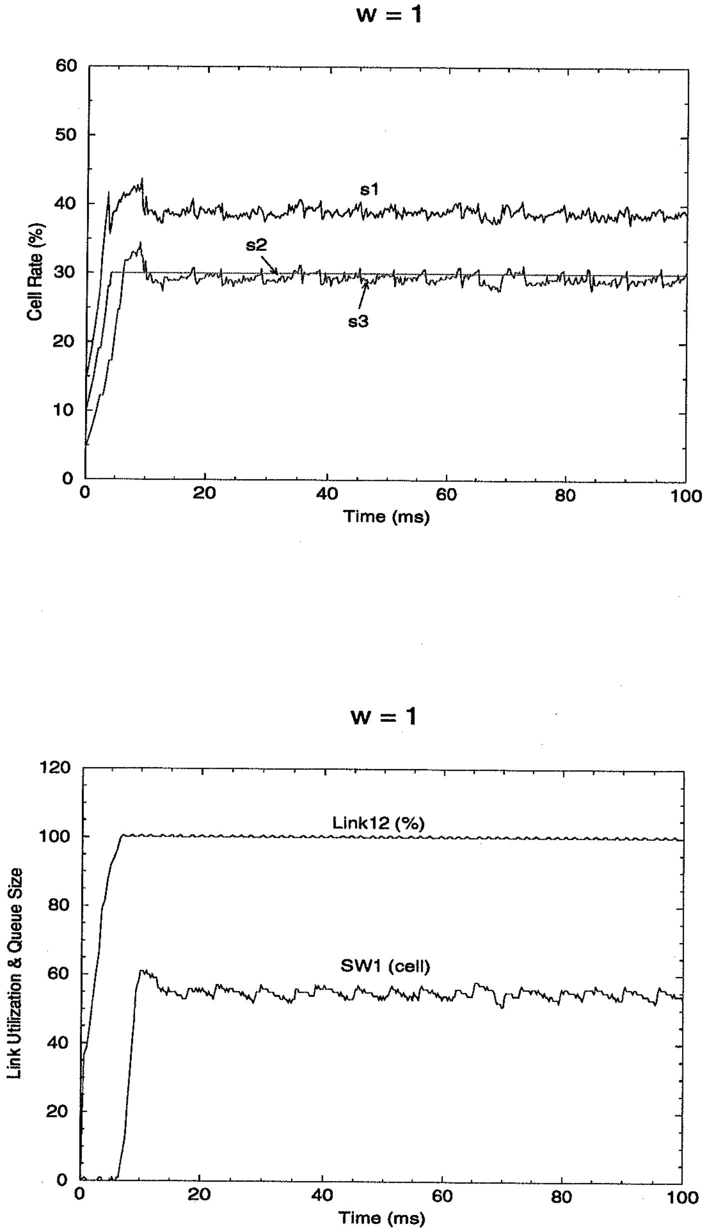


Figure 6: The cell rates of all connections, the link utilization and the queue size of the congested switch in the peer-to-peer network configuration; $w = 1$ for each session in the WMM policy.

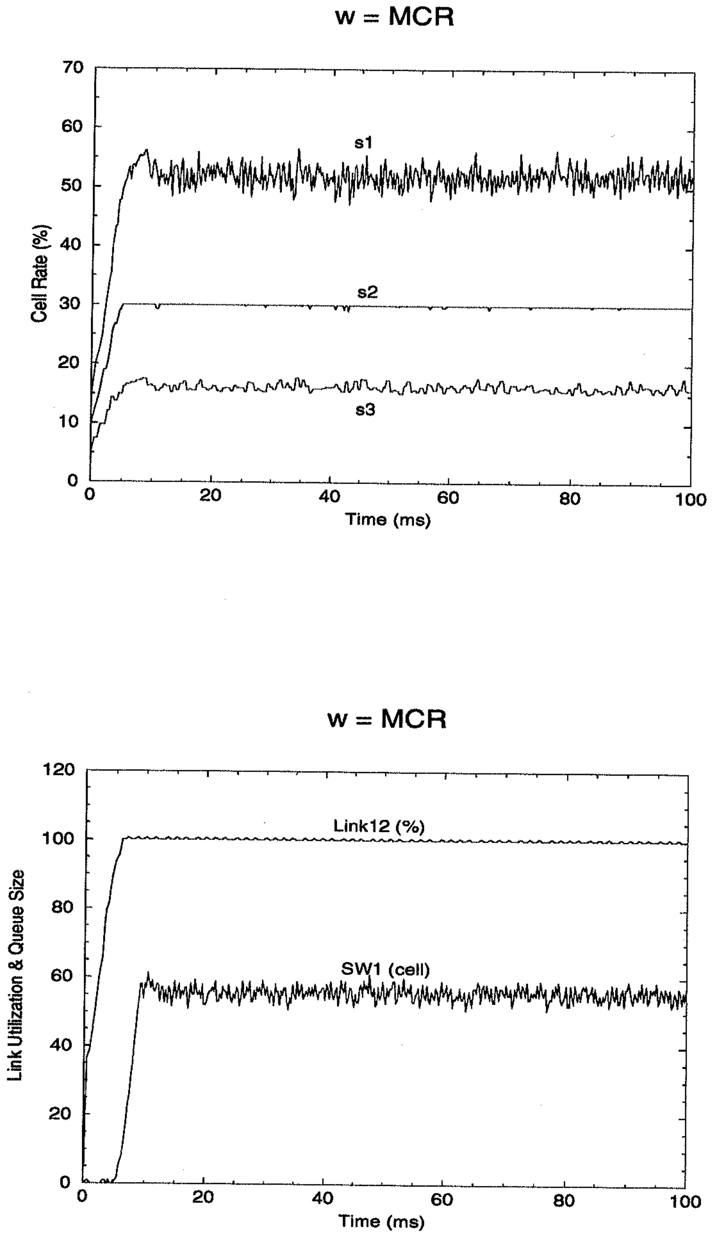
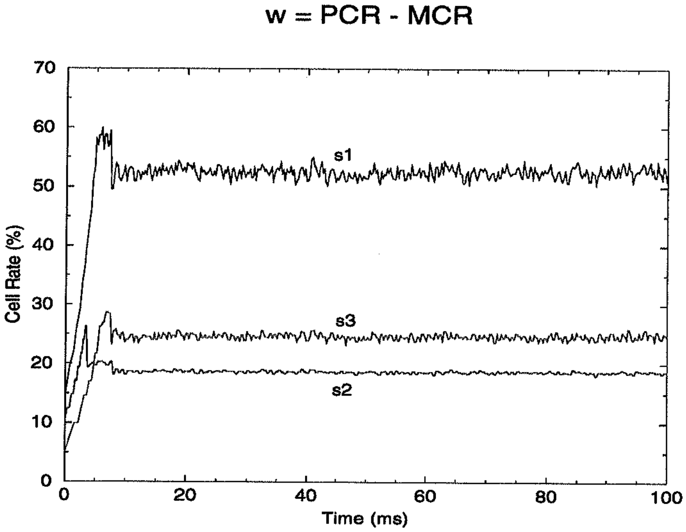


Figure 7: The cell rates of all connections, the link utilization and the queue size of the congested switch in the peer-to-peer network configuration; w = MCR for each session in the WMM policy.



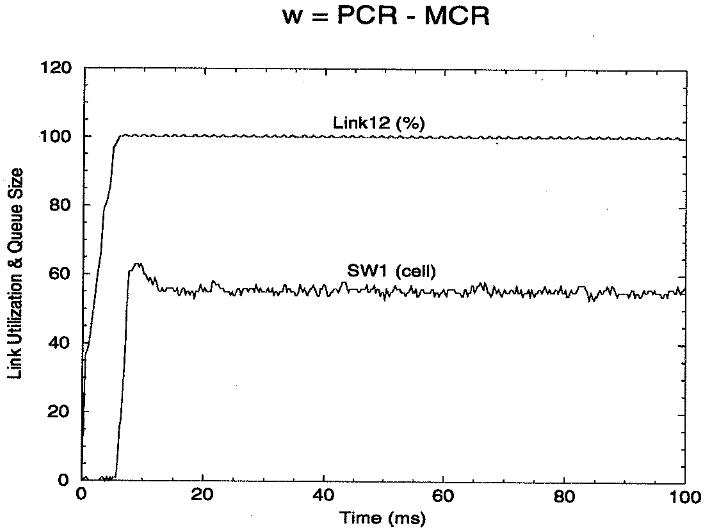


Figure 8: The cell rates of all connections, the link utilization and the queue size of the congested switch in the peer-to-peer network configuration; $w = \text{PCR} - \text{MCR}$ for each session in the WMM policy.

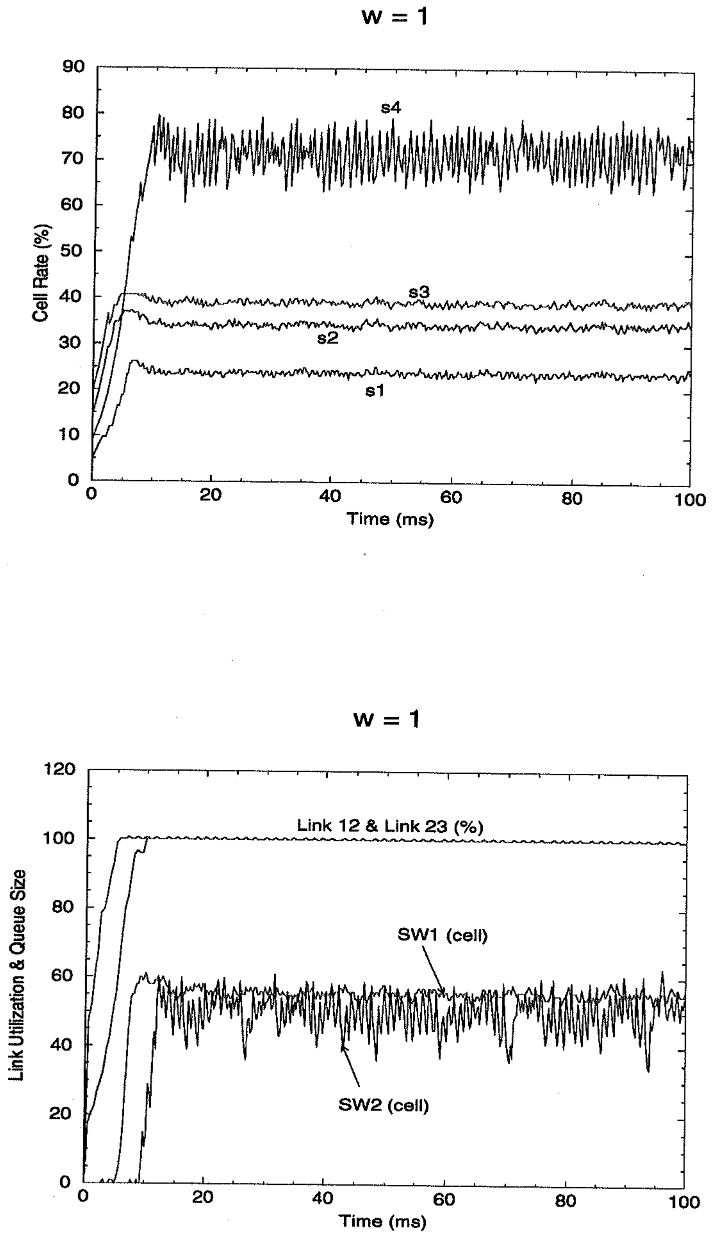
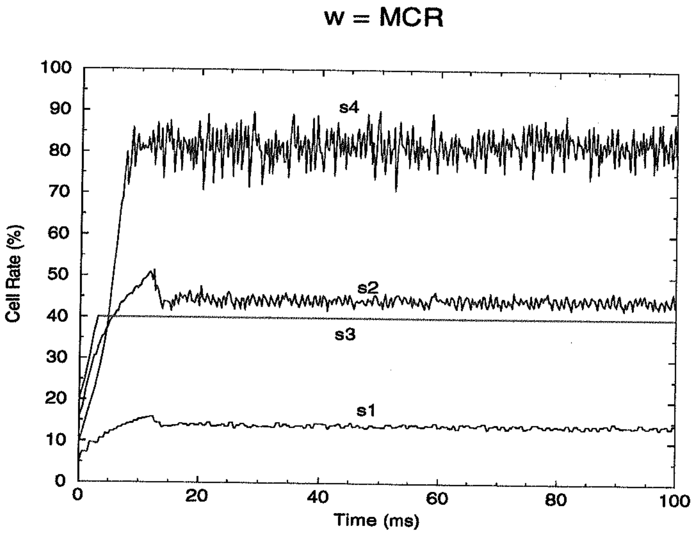


Figure 9: The cell rates of all connections, the link utilizations and the queue size of the congested switches in the three-node network configuration; $w = 1$ for each session in the WMM policy.



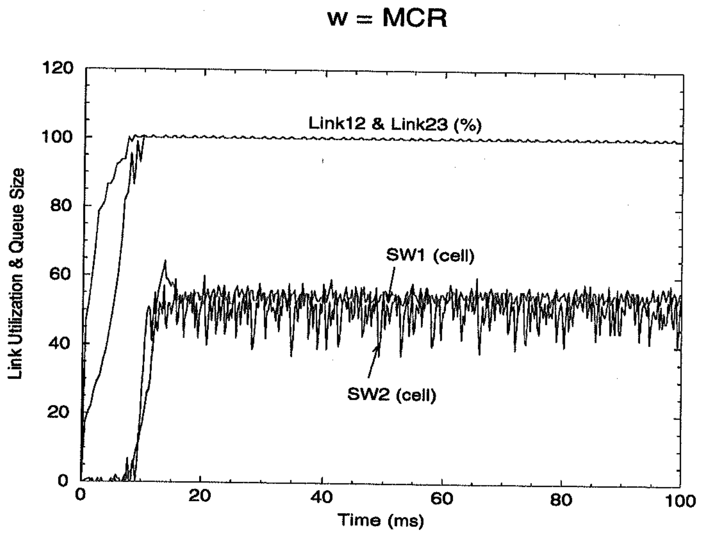
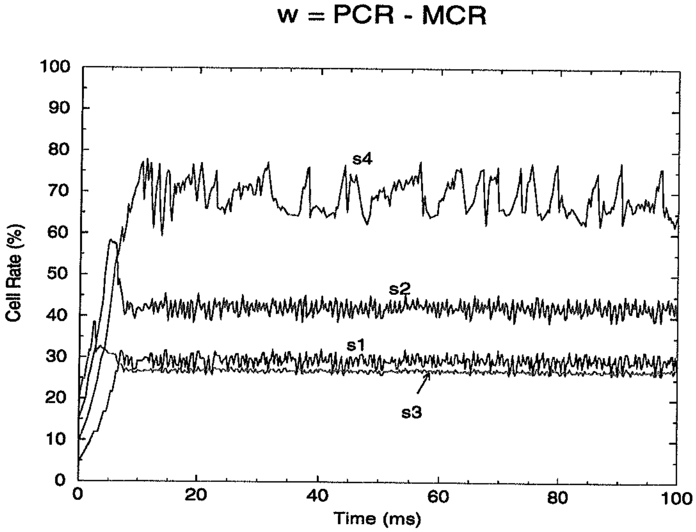


Figure 10: The cell rates of all connections, the link utilizations and the queue size of the congested switches in the three-node network configuration; w = MCR for each session in the WMM policy.



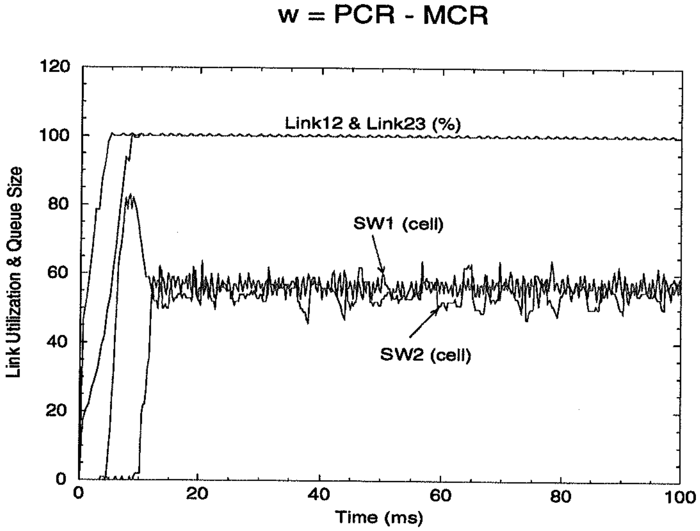
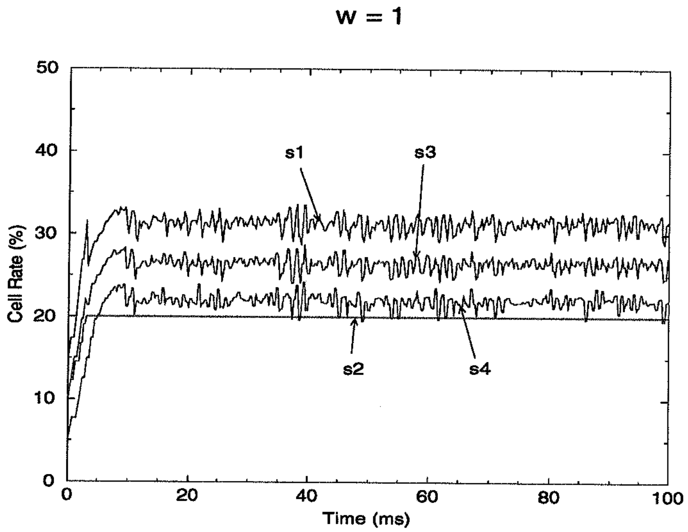


Figure 11: The cell rates of all connections, the link utilizations and the queue size of the congested switches in the three-node network configuration; $w = \text{PCR} - \text{MCR}$ for each session in the WMM policy.



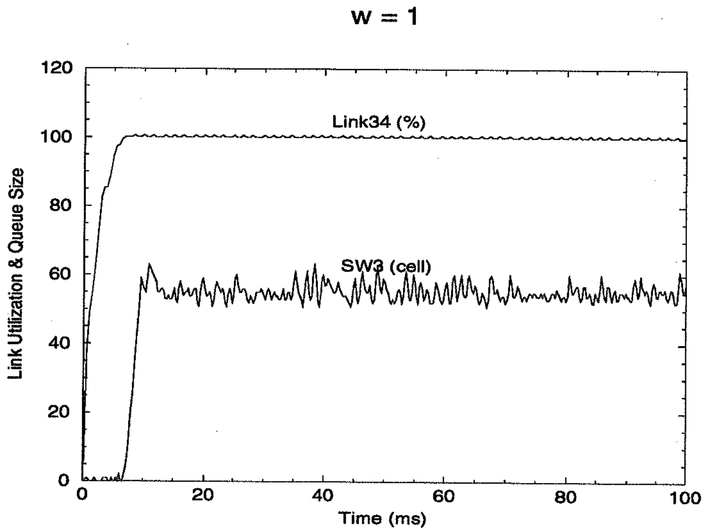


Figure 12: The cell rates of all connections, the link utilization and the queue size of the congested switch (SW3) in the parking lot network configuration; $w = 1$ for each session in the WMM policy.

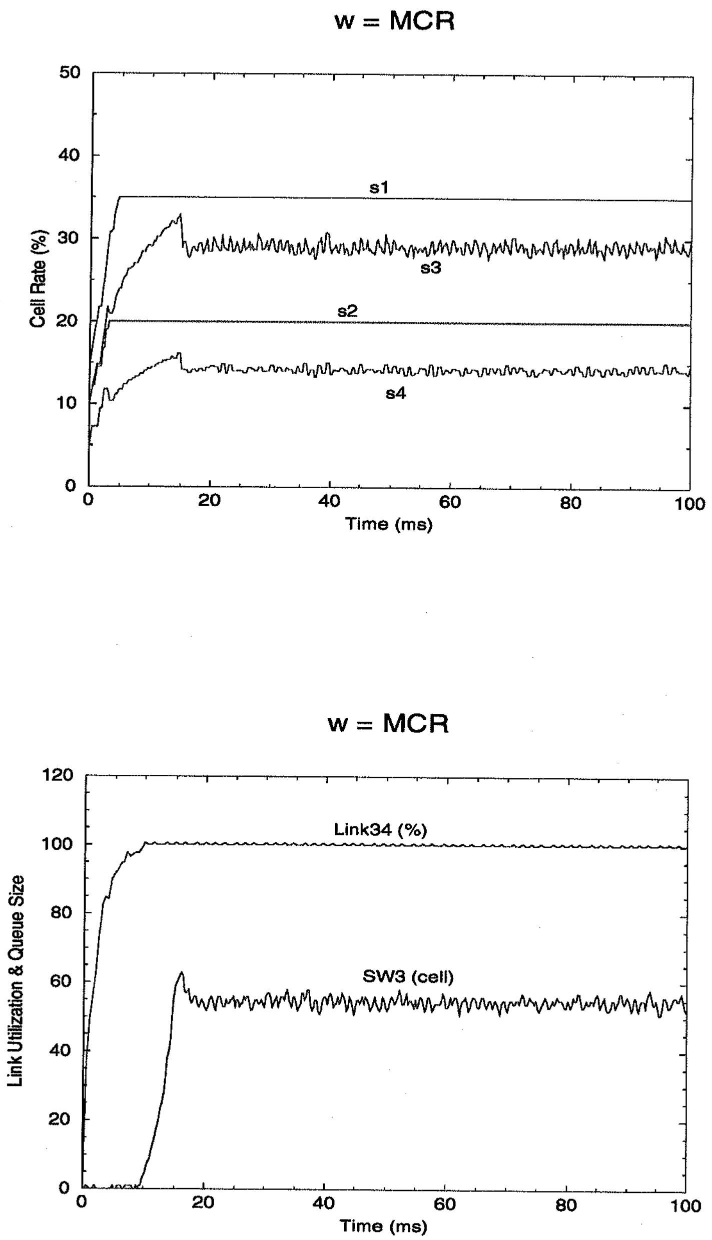


Figure 13: The cell rates of all connections, the link utilization and the queue size of the congested switch (SW3) in the parking lot network configuration; w = MCR for each session in the WMM policy.

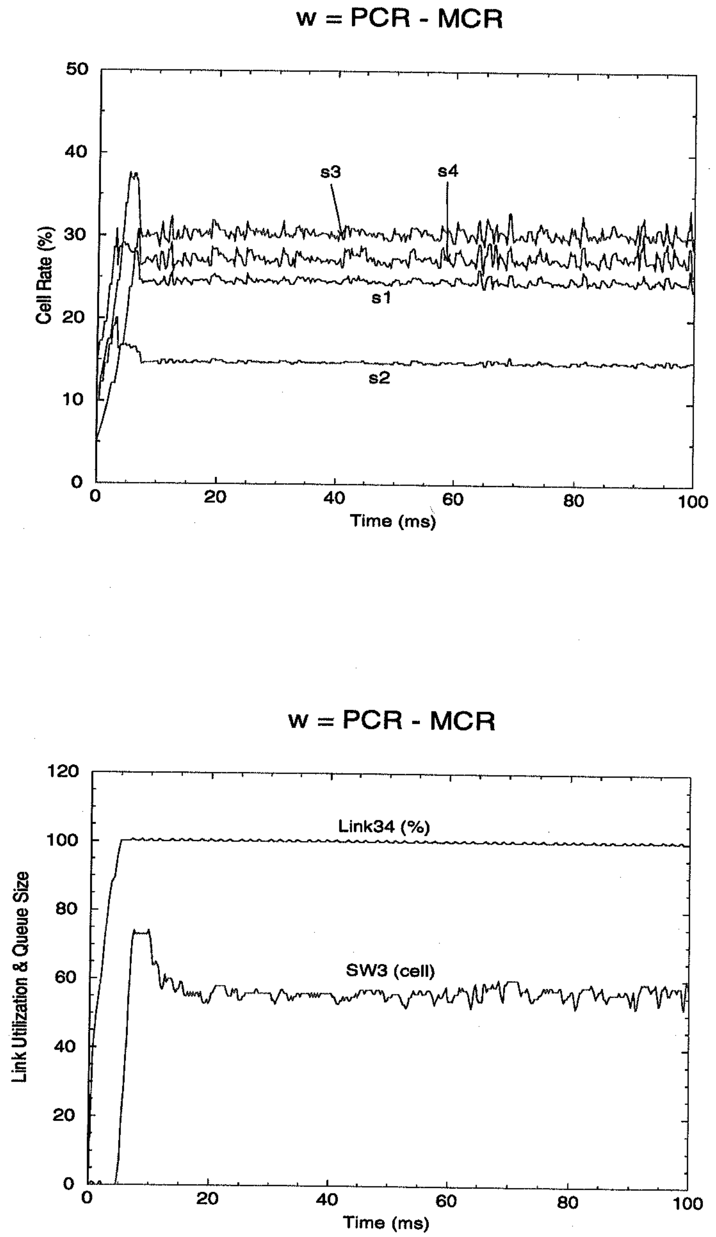


Figure 14: The cell rates of all connections, the link utilization and the queue size of the congested switch (SW3) in the parking lot network configuration; $w = \text{PCR} - \text{MCR}$ for each session in the WMM policy.

Appendix

Proof of Theorem 1:

To show the *"only if"* part, suppose that the ABR-feasible rate vector r is WMM and assume that on the contrary that there exists some session $s \in \mathcal{S}$ which has neither a WMM-bottleneck link with respect to r nor a rate assignment equal to its PCR. Then, for every saturated link ℓ ($F_\ell = C_\ell$) traversed by s, there must exist a session $p \neq s$ such that $\frac{r_p - \text{MCR}_p}{w_p} > \frac{r_s - \text{MCR}_s}{w_s}$. Since $r_s - \text{MCR}_s \geq 0$ due to ABR-feasibility, we have $r_p - \text{MCR}_p > 0$, or $r_p > \text{MCR}_p$. Thus the quantity

$$\delta_\ell = \begin{cases} \min\{(C_\ell - F_\ell),\ (\text{PCR}_s - r_s)\} & \text{if } F_\ell < C_\ell; \\ \min\{(r_p - \text{MCR}_p),\ (\text{PCR}_s - r_s)\} & \text{if } F_\ell = C_\ell. \end{cases}$$

is positive on all links ℓ traversed by s. Now let δ be the minimum of δ_ℓ over all links ℓ traversed by s. Therefore, we can increase r_s by δ while decreasing the same amount of rate from session r_p on the links ℓ traversed by s with $F_\ell = C_\ell$. We maintain ABR-feasibility without decreasing the rate of any session t with $\frac{r_t - \text{MCR}_t}{w_t} \leq \frac{r_s - \text{MCR}_s}{w_s}$ and this contradicts the WMM definition of rate vector r.

For the proof of the *"if"* part of Theorem 1, we assume that each session $s \in \mathcal{S}$ has either a WMM-bottleneck link with respect to the ABR-feasible rate vector r or a rate assignment equal to its PCR.

- *Case 1:* To increase the rate of any session s with $r_s < \text{PCR}_s$ while maintaining ABR-feasibility, we must decrease the rate of some session t traversing the WMM-bottleneck link ℓ of s (session s must go through a WMM-bottleneck link since $r_s < \text{PCR}_s$ and we have $F_\ell = C_\ell$ by the definition of a WMM-bottleneck link). Since $\frac{r_s - \text{MCR}_s}{w_s} \geq \frac{r_t - \text{MCR}_t}{w_t}$ for all t traversing ℓ (including those sessions with PCR assignments) by the definition of a WMM-bottleneck link, the rate assignment for any session s with $r_s < \text{PCR}_s$ satisfies the requirement for WMM policy.

- *Case 2:* For any session s with $r_s = \text{PCR}_s$, we cannot further increase the rate of r_s while maintaining ABR-feasibility. That is, we cannot generate another ABR-feasible rate vector $\hat{r}$ with $\hat{r}_s > r_s$. Thus, the rate assignment for any session s with $r_s = \text{PCR}_s$ satisfies the requirement for WMM.

Combining Cases 1 and 2, the rate vector r is WMM. $\square$

Correctness Proof of Algorithm 1:

Since 1) At least either one additional link is saturated or one session reaches its PCR at the end of each iteration (see Steps 1, 2 and 3); 2) At least one additional session is removed at the end of each iteration (Steps 1 to 6); and 3) The number of sessions in the network is a constant equal to $|\mathcal{S}|$; the algorithm terminates in at most $|\mathcal{S}|$ iterations. The correctness of this algorithm is proved by showing that each session will have either some WMM-bottleneck link or a rate assignment equal to its PCR when the algorithm terminates. Initially, the rate allocation (r_s^0) of each session $s \in \mathcal{S}$ is MCR_s. During each iteration, a rate increment of equal normalized amount (with respect to each connection's weight scale) is added to each session that has not yet passed through a saturated link or reached their PCRs. Thus, at the k^{th} iteration, we have $\frac{r_s-\mathrm{MCR}_s}{w_s} = \frac{r_t-\mathrm{MCR}_t}{w_t}$ for any sessions $s, t \in \mathcal{S}^k$.

- *Case 1:* Suppose that a link ℓ saturates at the k^{th} iteration, and $s \in \mathcal{S}^k$ traverses ℓ. We have $\frac{r_s-\mathrm{MCR}_s}{w_s} \geq \frac{r_t-\mathrm{MCR}_t}{w_t}$ for every $t \in \mathcal{S}$ traversing ℓ (including such session t with its PCR assignment). That is, link ℓ is a WMM-bottleneck link for session s.
- *Case 2:* Suppose that session s reaches its PCR at the k^{th} iteration. Then $r_s^k = \mathrm{PCR}_s$ and r_s^k will be not be increased further during future iterations.

As a result, upon termination of the algorithm, each session either has some WMM-bottleneck link or a rate assignment equal to its PCR. By Theorem 1, the final rate vector r satisfies the WMM policy. $\square$

References

[1] The ATM Forum Technical Committee, "Traffic Management Specification, Version 4.0," *ATM Forum Contribution, AF-TM 96-0056.00,* April 1996.

[2] D. Bertsekas and R. Gallager, *Data Networks,* Chapter 6, Prentice Hall, 1992.

[3] A. Charny, D. Clark, and R. Jain, "Congestion Control with Explicit Rate Indication," *Proc. IEEE ICC'95,* pp. 1954-1963, June 1995.

[4] E. M. Gafni, "The Integration of Routing and Flow Control for Voice and Data in a Computer Communication Network," *Ph.D. Thesis,* Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, August 1982.

[5] H. P. Hayden, "Voice Flow Control in Integrated Packet Networks," *M.S. Thesis,* Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, June 1981.

[6] A. Heybey, "The Network Simulator – Version 2.1," *Laboratory of Computer Science,* MIT, September 1990.

[7] Y. T. Hou, H. Tzeng, and S. S. Panwar, "A Simple ABR Switch Algorithm For the Weighted Max-Min Fairness Policy," *Proc. IEEE ATM'97 Workshop,* pp. 329-338, Lisbon, Portugal, May 25-28, 1997.

[8] Y. T. Hou, H. Tzeng, S. S. Panwar, and V. P. Kumar, "Fair Network Bandwidth Allocation with Minimum Rate Guarantee and Its ABR Implementations," *Proc. IEEE International Conference on Communications (ICC '97),* pp. 1411-1416, Montreal, Quebec, Canada, June 8-12, 1997.

[9] Y. T. Hou, H. Tzeng, and S. S. Panwar, "A Generic Weight Based Network Capacity Sharing Policy for ATM ABR Service," *Tech. Report CATT 97-118,* Center for Advanced Technology in Telecommunications (CATT), Polytechnic University, Brooklyn, NY, July 1, 1997.

[10] Y. T. Hou, H. Tzeng, and S. S. Panwar, "A Weighted Max-Min Fair Rate Allocation for Available Bit Rate Service," *Proc. IEEE Global Communications Conference (GLOBECOM'97),* Phoenix, AZ, USA, Nov. 4-8, 1997.

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

[12] J. M. Jaffe, "Bottleneck Flow Control," *IEEE Trans. on Comm.,* vol. COM-29, no. 7, pp. 954-962, July 1981.

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

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

[15] 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),* pp. 143-154, Sept. 1995.

[16] J. Mosely, "Asynchronous Distributed Flow Control Algorithms," *Ph.D. Thesis,* Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, June 1984.

[17] K. K. Ramakrishnan, R. Jain, and D.-M. Chiu, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer – Part IV: A Selective Binary Feedback Scheme for General Topologies Methodology," *DEC-TR-510,* Digital Equipment Corporation, 1987.

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

[19] K.-Y. Siu and H.-Y. Tzeng, "Intelligent Congestion Control for ABR Service in ATM Networks," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 5, pp. 81-106, October 1994.

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

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

[22] N. Yin and M. G. Hluchyj, "On Closed-Loop Rate Control for ATM Cell Relay Networks," *Proc. IEEE INFOCOM '94*, pp. 99-108, June 1994.

[23] 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.