A Weighted Max-Min Fair Rate Allocation for Available Bit Rate Service *

Yiwei Thomas Hou,† Henry H.-Y. Tzeng,‡ Shivendra S. Panwar§

Abstract

An important concept in the ABR service model is the minimum cell rate (MCR) guarantee as well as the peak cell rate (PCR) constraint for each connection. Because of the MCR and PCR constraints, the classical max-min policy no longer suffices to determine rate allocation since it does not support either the MCR or the PCR.

To support the MCR/PCR constraints for each connection, we present the *Weighted Max-Min (WMM)* policy, with the "weight" of each connection being its MCR requirement (we assume a nonzero MCR for each ABR connection). Furthermore, an explicit-rate (ER) based ABR switch algorithm is developed to achieve the WMM policy in the distributed network environment. Our ABR algorithm is proven to converge to the WMM policy through distributed and asynchronous iterations. The performance of our ABR algorithm is demonstrated by simulation results based on the benchmark network configurations suggested by the ATM Forum.

1 Introduction

A key performance issue associated with the ABR service is fair allocation of network bandwidth for each virtual connection (VC). The ATM Forum has adopted the max-min policy to allocate network bandwidth for ABR service [2]. Many efforts to design distributed ABR algorithms to achieve the max-min policy have been made [3, 9, 10, 11, 12].

There are several issues associated with the classical max-min policy. For example, the max-min policy does not address how to support each connection's minimum rate requirement (a bandwidth QoS feature offered to ABR traffic by ATM networks) and peak rate constraint (imposed by the host application or terminal equipment). Also, it does not offer flexible pricing criterion for network providers.

In this paper, we present the weighted max-min (WMM) policy with both MCR/PCR support for each connection. We let the "weight" of an ABR connection be identical to its MCR requirement.[1] This policy may reflect, for example, a real or nominal pricing proportional to the MCR of each connection.

The WMM policy was first informally described in [7, 13] for the simple single node case without the PCR constraint. In our previous work [5], we formally defined this policy with MCR guarantee but without the PCR constraint. In this paper, we further generalize this policy with a PCR constraint for each connection.

The centralized algorithm for the WMM policy requires global information and thus is difficult to maintain in real world networks. Therefore, we develop a distributed algorithm consistent with the ATM Forum ABR Traffic Management framework to achieve this policy. Our ABR algorithm is motivated by the *Consistent Marking* technique by Charny [3], which achieves the classical max-min policy without MCR/PCR constraints. We extend this technique and design a distributed algorithm to achieve the WMM policy with MCR/PCR support.

The remainder of this paper is organized as follows. In Section 2, we define the WMM policy. In Section 3, we develop a distributed algorithm using the ABR protocol to achieve the WMM policy and give a proof of its convergence. In Section 4, we present simulation results of our ABR algorithm on a few benchmark network configurations suggested by the ATM Forum. Section 5 concludes this paper and points out future research directions.

2 The MCR-Weighted Max-Min Rate Allocation Policy

In our model, a network $\mathcal{N}$ is characterized by a set of links $\mathcal{L}$ and sessions $\mathcal{S}$.[2] 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 MCR requirement and PCR constraint for session $s \in \mathcal{S}$. 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

*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. 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.

[1]Strictly speaking, this policy does not work for a VC with zero MCR requirement. But in a high speed ATM network environment, the assignment of a small MCR value to a VC shall not pose any fundamental technical difficulty. Therefore, we assume a nonzero MCR for each VC throughout our paper.

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

capacity, i.e.

$$\sum_{\text{all } s \in \mathcal{S} \text{ traversing } \ell} \text{MCR}_s \leq C_\ell \quad \text{for every } \ell \in \mathcal{L}.$$

□

This assumption is enforced by admission control at call setup time to determine whether or not to accept a new connection.

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

$$\text{MCR}_s \leq r_s \leq \text{PCR}_s \quad \text{for all } s \in \mathcal{S};$$
$$F_\ell \leq C_\ell \quad \text{for all } \ell \in \mathcal{L}.$$

Informally, the weighted max-min policy achieves max-min for the normalized (with respect to each individual connection's MCR) rate vector r. Formally, this policy is defined as follows.

Definition 1 A rate vector r is *weighted 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 $\frac{r_s}{\text{MCR}_s} \geq \frac{r_t}{\text{MCR}_t}$ and $r_t > \hat{r}_t$. □

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

Definition 2 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 $\frac{r_s}{\text{MCR}_s} \geq \frac{r_t}{\text{MCR}_t}$ for all sessions t traversing link ℓ. □

It can be shown that the following two theorems hold for the WMM policy [6].

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. □

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

The following centralized algorithm computes the rate allocation for each session in any network $\mathcal{N}$ such that the WMM policy is satisfied.

Algorithm 1 A Centralized Algorithm

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

The following example illustrates the WMM rate allocation for a simple peer-to-peer network configuration using the above centralized algorithm.

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. Assume that all links are of unit capacity. The MCR requirement and PCR constraint for each session are listed in Table 1. Using the centralized algorithm for the WMM policy (Algorithm 1), we obtain the WMM rate allocation for each session in Table 1. □

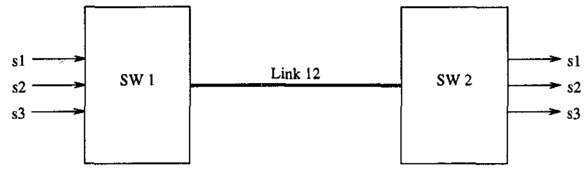

Figure 1: The peer-to-peer network configuration.

Session	MCR	PCR	WMM Rate Allocation
$s1$	0.15	1.00	0.525
$s2$	0.10	0.30	0.30
$s3$	0.05	0.50	0.175

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

As shown above, the centralized algorithm for the WMM policy requires global information, which is difficult to maintain in real world networks. To achieve the WMM policy in a distributed network environment, we will develop a distributed algorithm using the ABR flow control protocol in the next section.

3 A Distributed ABR Implementation

A generic rate-based closed-loop congestion 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), minimum cell rate (MCR) requirement, explicit rate (ER), congestion indication (CI) bit and no increase (NI) bit.

Our distributed implementation for the WMM policy employs ER calculation and is based on the *Consistent Marking* technique by Charny [3], which was designed to achieve the classical max-min policy. It is a powerful mechanism to bring network bandwidth allocation to max-min rates through distributed and asynchronous iterations and can be generalized for the design of a broad class of distributed algorithms for other rate allocation policies. In the following, we generalize the Consistent Marking technique for our WMM policy with MCR/PCR support.

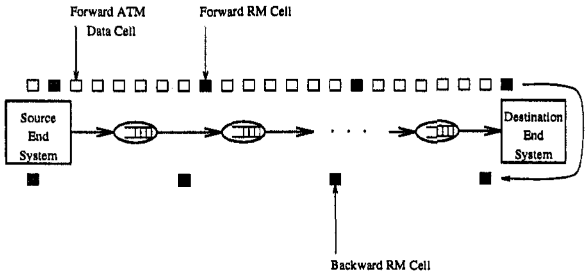

Figure 2: Rate-based closed-loop flow control for an ABR virtual connection.

3.1 The Protocol

We first specify the source behavior of our ABR algorithm [1].

Algorithm 2 Source Behavior

A 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) cell from the destination, the ACR at the source is adjusted to:
ACR := ER. □

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

At each output port of a switch, we maintain a table and keep track of the state information of each traversing VC (so-called per-VC accounting). Specifically, for each RM cell traversing a link, the switch records the CCR and MCR for each VC and performs the switch algorithm (Algorithm 4) at this link. Each link $\ell \in \mathcal{L}$ also maintains a variable, μ_ℓ, to estimate the MCR-normalized WMM-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 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_ℓ^i: CCR value of session $i \in \mathcal{G}_\ell$ at link ℓ.

MCR^i: MCR requirement of session i.

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

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

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

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

μ_ℓ: MCR-normalized advertised rate at link ℓ, calculated as follows:

Algorithm 3 μ_ℓ Calculation

$$\mu_\ell := \begin{cases} \infty & \text{if } n_\ell = 0;^3 \\ \frac{C_\ell - \sum_{i\in\mathcal{G}_\ell} r_\ell^i}{\sum_{i\in\mathcal{G}_\ell} \mathrm{MCR}^i} + \max_{i\in\mathcal{G}_\ell} \frac{r_\ell^i}{\mathrm{MCR}^i} & \text{if } n_\ell = |\mathcal{Y}_\ell|; \\ \frac{C_\ell - \sum_{i\in\mathcal{Y}_\ell} r_\ell^i}{\sum_{i\in\mathcal{U}_\ell} \mathrm{MCR}^i} & \text{otherwise.} \end{cases}$$

The following algorithm specifies our switch behavior, which is initialized with: $\mathcal{G}_\ell = \emptyset$; $n_\ell = 0$; $\mu_\ell = \infty$.

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 exit⁴{
        G_ℓ := G_ℓ − {i};    n_ℓ := n_ℓ − 1;
        table_update();
        }
    if RM cell signals session initiation {
        G_ℓ := G_ℓ ∪ {i};    n_ℓ := n_ℓ + 1;
        r_ℓ^i := CCR;   MCR^i := MCR;   b_ℓ^i := 0;
        table_update();
        }
    else /* i.e. RM cell belongs to an ongoing active
        session */ {
        r_ℓ^i := CCR;
        if (r_ℓ^i / MCR^i ≤ μ_ℓ) then b_ℓ^i := 1;
        table_update();
        }

    Forward RM(CCR, MCR, ER) towards its
    destination;
}


Upon the receipt of a backward RM(CCR, MCR, ER)
cell from the destination of session i {
    ER := max{min{ER, μ_ℓ · MCR}, MCR};
    Forward RM(CCR, MCR, ER) towards its source;
}

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

[3] In fact, μ_ℓ can be set to any constant when $n_\ell = 0$.

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

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

rate_calculation_2: use Algorithm 3 to calculate μ_ℓ;

if $(\mu_\ell < \mu_\ell^1)$, then {
Unmark any marked session $i \in \mathcal{G}_\ell$ at link ℓ with $\frac{r_\ell^i}{\text{MCR}^i} > \mu_\ell$;
rate_calculation_3: use Algorithm 3 to calculate μ_ℓ again;
}[5]

} □

By the operations of Algorithms 2 and 4, we have the following fact for the ACR at the source and the CCR in the RM cell.

Fact 1 For every ABR 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 $\text{MCR}^s \leq \text{CCR}^s \leq \text{PCR}^s$. □

In the following, we give a sketch of the proof that rate allocation by the above distributed ABR algorithm converge to the WMM policy through distributed and asynchronous iterations. For readers who are interested in the details of the complete proof, please see [6].

3.2 Convergence of Distributed Rate Allocation

We give the following definition for *marking-consistent*.

Definition 3 Let $\mathcal{Y}_\ell$ be the set of sessions that are marked at link $\ell \in \mathcal{L}$ and μ_ℓ be calculated according to Algorithm 3. The marking of sessions at link $\ell \in \mathcal{L}$ is *marking-consistent* if $\frac{r_\ell^i}{\text{MCR}^i} \leq \mu_\ell$ for every session $i \in \mathcal{Y}_\ell$. □

The following key lemma shows the marking property at a link when the switch algorithm is performed for a traversing RM cell.

Lemma 1 After the switch algorithm is performed for each RM cell traversing a link, the marking of sessions at this link is marking-consistent. □

Let M be the total number of iterations needed to execute Algorithm 1, $M \leq |\mathcal{S}|$. Let $\mathcal{S}_i$, $1 \leq i \leq M$ be the set of sessions being removed at the end of the ith iteration, i.e. sessions in $\mathcal{S}_i$ have either reached their WMM-bottleneck link rate or their PCRs during the ith iteration of Algorithm 1. Let $\mathcal{L}_i$, $1 \leq i \leq M$ be the set of links traversed by sessions in $\mathcal{S}_i$. Let τ_i, $1 \leq i \leq M$ be defined as following:

$$\tau_i = \frac{r^s}{\text{MCR}^s} \quad \text{for every } s \in \mathcal{S}_i, \ 1 \leq i \leq M,$$

where r^s is the final WMM rate allocation for session s by Algorithm 1. By Assumption 1, we have $\tau_1 \geq 1$. By the operation of Algorithm 1, for a session which has not yet gone through a saturated link or reached its PCR, its rate allocation normalized with respect to its MCR increases at each iteration. Therefore, we have the following property for τ_i, $1 \leq i \leq M$,

$$1 \leq \tau_1 < \tau_2 < \ldots < \tau_M.$$

Lemma 2 There exists a $T_1 \geq 0$ such that:

i) If $\tau_1 = \frac{C_\ell}{\sum_{i \in \mathcal{G}_\ell} \text{MCR}^i} \leq \frac{\text{PCR}^s}{\text{MCR}^s}$ for $s \in \mathcal{S}_1$, i.e., the WMM-bottleneck link rate is reached before some session $s \in \mathcal{S}_1$ reaches its PCR, then for $t \geq T_1$, the following statements hold.

1. $\mu_\ell = \tau_1$ for every link $\ell \in \mathcal{L}_1$.
2. The ER field of every returning RM cell of session $i \in \mathcal{S}_1$ satisfies $\text{ER} = \tau_1 \cdot \text{MCR}^i$.
3. The ACR at source for every session $i \in \mathcal{S}_1$ satisfies $\text{ACR} = \tau_1 \cdot \text{MCR}^i$.
4. $b_\ell^i = 1$, $r_\ell^i = \tau_1 \cdot \text{MCR}^i$ for every session $i \in \mathcal{S}_1$ and every link ℓ traversed by session $i \in \mathcal{S}_1$.
5. The ER field of every returning RM cell of session $j \in (\mathcal{S} - \mathcal{S}_1)$ satisfies $\text{ER} > \tau_1 \cdot \text{MCR}^j$.
6. The ACR at source for every session $j \in (\mathcal{S} - \mathcal{S}_1)$ satisfies $\text{ACR} > \tau_1 \cdot \text{MCR}^j$.
7. The recorded CCR of session $j \in (\mathcal{S} - \mathcal{S}_1)$ satisfies $r_\ell^j > \tau_1 \cdot \text{MCR}^j$ at every link ℓ traversed by session j.

ii) If $\tau_1 = \frac{\text{PCR}^s}{\text{MCR}^s} < \frac{C_\ell}{\sum_{i \in \mathcal{G}_\ell} \text{MCR}^i}$ for $s \in \mathcal{S}_1$, i.e., some session $s \in \mathcal{S}_1$ reaches its PCR before the WMM-bottleneck link rate is reached, then for $t \geq T_1$, the following statements hold.

1. $\mu_\ell > \tau_1$ for every link $\ell \in \mathcal{L}_1$.
2. The ER field of every returning RM cell of session $i \in \mathcal{S}_1$ satisfies $\text{ER} = \text{PCR}^i$.
3. The ACR at source for every session $i \in \mathcal{S}_1$ satisfies $\text{ACR} = \text{PCR}^i$.
4. $b_\ell^i = 1$, $r_\ell^i = \text{PCR}^i$ for every session $i \in \mathcal{S}_1$ and every link ℓ traversed by session $i \in \mathcal{S}_1$.
5. — 7. Same as statements i)–5 to i)–7, respectively. □

[5] Both μ_ℓ^1 and μ_ℓ follow the same μ_ℓ calculation in Algorithm 3. In most cases, μ_ℓ calculated by rate_calculation_2 is greater than or equal to μ_ℓ^1 and rate_calculation_3 is not used. A unique case where μ_ℓ calculated by rate_calculation_2 is less than μ_ℓ^1 and another around of unmarking and rate_calculation_3 is necessary is given in [6].

Lemma 2 is used as the base case for induction on the index i of $\mathcal{S}_i$, $1 \leq i \leq M-1$. That is, it can be shown that once the rate allocation and session marking for sessions in $\mathcal{S}_i$ have reached the targeted rates, then the target rate allocation and marking property will hold for sessions in $\mathcal{S}_{i+1}$ [6].

Theorem 3 After the number of active sessions in the network stabilizes, the rate allocation for each session by the ABR algorithm converges to the WMM policy. □

4 Simulation Results

In this section, we implement our distributed algorithm on our network simulator [4] and perform simulations on a few benchmark network configurations suggested by the ATM Forum Traffic Management Group. The purpose of our work in this section is to have some quantitative insights on the convergence time of our ABR algorithm.

The network configurations that we use are the peer-to-peer (Fig. 1) and the *parking-lot* (Fig. 4) configurations. The ATM switches are assumed to have output port buffers with a speedup equal to the number of their ports. The buffer of each output port of a switch employs the simple FIFO queueing discipline and is shared by all VCs going through that port. At each output port of an ATM switch, we implement our ABR switch algorithm.

Table 2 lists the parameters used in our simulation. The link capacity is 150 Mbps. 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 the ER calculation. The distance from source/destination to the switch is 1 km and the link distance between ATM switches is 1000 km (corresponding to a wide area network) and we assume that the propagation delay is 5 μs per km.

End System	PCR	PCR
	MCR	MCR
	ICR	MCR
	Nrm	32
Link	Speed	150 Mbps
	C_ℓ	142.5 Mbps
Switch	Cell Switching Delay	4 μs

Table 2: Simulation parameters.

The Peer-to-Peer Network Configuration

For this configuration (Fig. 1), the output port link of SW1 is the only potential bottleneck link for all sessions. The specific MCR requirement, PCR constraint, and WMM rate allocation is given in Example 1.

Fig. 3 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). Each session starts with its MCR. The first RM cell for each session returns to the source after one round trip time (RTT), or 10 ms. After a transient period, we see that the cell rate of each session converges to the final rate listed in Table 1. Also, we find that during the course of distributed iterations, the ACR of each session maintains ABR-feasibility, i.e., MCR $\leq$ ACR $\leq$ PCR. Here the RTT is 10 ms and it takes less than 15 ms for our ABR algorithm to converge to the final rates.

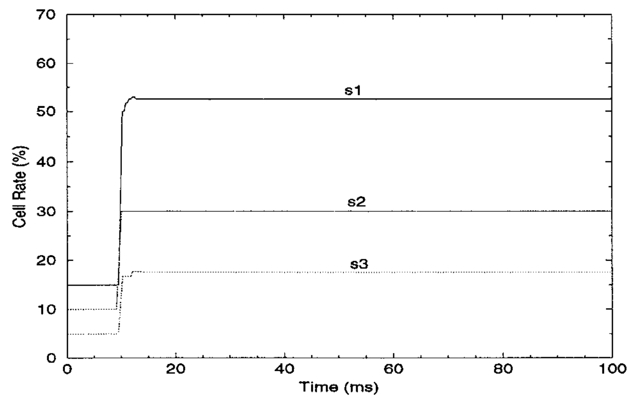

Figure 3: The cell rates of all connections for the WMM policy in the peer-to-peer network configuration.

The Parking Lot Network Configuration

The specific parking lot configuration that we use is shown in Fig. 4 where sessions $s1$ and $s2$ start from the first switch and go to the last switch [8]. Sessions $s3$ and $s4$ start from SW2 and SW3, respectively, and terminate at the last switch.

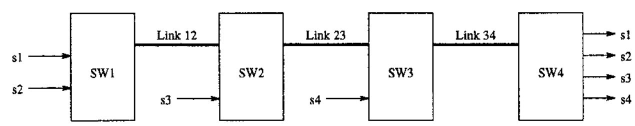

Figure 4: The parking-lot network configuration.

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

Session	MCR	PCR	WMM Rate Allocation
$s1$	0.15	0.35	0.35
$s2$	0.10	0.20	0.20
$s3$	0.10	0.50	0.30
$s4$	0.05	0.50	0.15

Table 3: WMM rate allocation for parking-lot network configuration.

Fig. 5 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. Here the maximum RTT among all sessions is 30 ms ($s1$ and $s2$) and it takes our ABR algorithm less than 2 RTT to converge to the final optimal rates.

In summary, based on the simulation results in this section, we have demonstrated that our distributed ABR algorithm achieves the WMM rate allocation policy with fast convergence time.

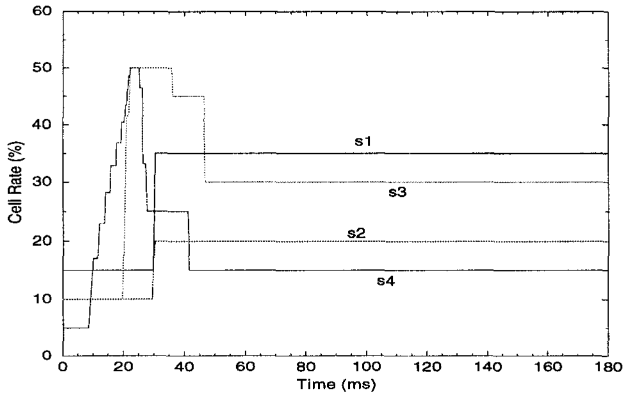

Figure 5: The cell rates of all connections for the WMM policy in the parking-lot network configuration.

5 Concluding Remarks

We have presented the MCR-weighted max-min rate allocation policy to support both the MCR requirement and PCR constraint for each ABR connection. Furthermore, we have developed a distributed algorithm in the context of the ATM Forum ABR traffic management framework to achieve the WMM policy. Our ABR algorithm is proven to converge to WMM rate allocation through distributed and asynchronous iterations. Simulation results based on benchmark network configurations demonstrate its fast convergence property.

Our future work will focus on other issues in our ABR implementation for the WMM policy. One challenging issue for us is to reduce the storage and computational complexity of our switch algorithm and yet be able to provide a rigorous proof of the algorithm's convergence. Other issues include system transient behavior, rate of convergence, and network buffer requirements, which are becoming increasingly important as ABR service is deployed for data communications in ATM networks.

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

[5] Y. T. Hou, H. Tzeng, and V. P. Kumar, "On Rate Allocation Policies with Minimum Cell Rate Guarantee for ABR Service in ATM Networks," *Proc. 15th International Teletraffic Congress (ITC-15),* pp. 1269-1280, Washington, DC, June 22-27, 1997.

[6] Y. T. Hou, H. Tzeng, and S. S. Panwar, "The Weighted Max-Min Fair Rate Allocation for ABR Service: Theory and Implementation," *Technical Report CATT 97-110,* Center for Advance Technology in Telecommunications (CATT), Polytechnic University, Brooklyn, NY, March 17, 1997.

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

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

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

[10] 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.

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

[12] 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.

[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.