A Generic Weight-Based Network Bandwidth Sharing Policy for ATM ABR Service *

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

Abstract

This paper presents a novel generic weight-based network bandwidth sharing policy and an available bit rate (ABR) algorithm that achieves this policy. Our policy supports the minimum cell rate (MCR) requirement and peak cell rate (PCR) constraint of each connection and allocates network bandwidth among all connections based on a weight associated with each connection. To achieve this policy for ABR connections, we design an ABR algorithm which employs per virtual connection (VC) accounting to keep track of the state information of each VC. Our ABR algorithm is proven to provide guaranteed convergence to our generic weight-based rate allocation policy under any network configuration and any set of link distances. Simulation results show that our ABR algorithm has a fast convergence property.

1 Introduction

A key performance issue associated with ABR service is the choice of a network bandwidth sharing policy among competing connections. The classical max-min policy has been suggested to allocate network bandwidth among ABR connections [1]. Informally, the max-min policy attempts to maximize the smallest rate among all connections; given the best smallest rate allocation, the next smallest rate allocation is maximized, etc. [3]. There are a few drawbacks associated with using the classical max-min policy for ABR service. First of all, the max-min policy, as it stands, cannot support the MCR/PCR constraints of each connection. Secondly, the max-min policy treats each connection with equal priority and thus is not flexible enough for network providers wishing to introduce differential service options to user connections.

Prior efforts to extend the classical max-min policy for ABR service include the so-called *MCRadd* policy [6, 9, 16] and the *MCRprop* policy [8, 9, 16]. Both policies first guarantee the minimum rate of each connection. Under MCRadd, the remaining network bandwidth is shared among all connections using the max-min policy, i.e., equal weight for all connections; while under MCRprop, the remaining bandwidth is shared among all connections using MCR-proportional max-min policy.

In this paper, we present a generic weight-based network bandwidth sharing policy, also called *Weight-based Max-Min (WMM)*, which generalizes the MCRadd and MCRprop policies in [6, 8, 9, 16]. We associate each connection with a generic weight, which is decoupled (or independent) from its minimum rate. Our policy supports MCR for each connection and allocates the remaining network bandwidth among all connections based on each connection's weight. Our policy offers a flexible service priority option to each user connection.

Our WMM policy sets up a network bandwidth allocation optimality criterion. A centralized algorithm for the WMM policy requires globally information, which cannot be applied directly to ABR service. To achieve our WMM policy for ABR service, which employs a distributed flow control mechanism, we need to design a distributed switch algorithm. Our ABR algorithm is motivated by the *Consistent Marking* technique in [5], which achieves the classical max-min. We extend this technique and design a distributed algorithm for our WMM policy with the support of a minimum rate requirement, a peak rate constraint, and a weight for each connection. Our ABR algorithm is proven to converge to our WMM policy through distributed and asynchronous iterations under any network configuration and any set of link distances. To show the convergence property of our ABR algorithm, we implement our ABR algorithm on a few benchmark network configurations suggested by the ATM Forum and use simulations to demonstrate its performance.

The remainder of this paper is organized as follows. In Section 2, we define our generic weight-based max-min (WMM) rate allocation policy. Section 3 presents an ABR algorithm that provides guaranteed convergence to our WMM policy. Simulation results of our ABR algorithm are presented in Section 4. Section 5 concludes this paper and points out future research directions.

---

*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), Polytechnic University, Brooklyn, NY.

†Y. T. Hou is currently with Fujitsu Labs of America, Santa Clara, CA. This work was performed while he was with the Dept. of Electrical Engineering, 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.

2 A Weight-Based Rate Allocation Policy

In our model, a network of ATM switches are interconnected by a set of links $\mathcal{L}$. A set of connections $\mathcal{S}$ traverses one or more links in $\mathcal{L}$ and each connection 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 $F_\ell = C_\ell$. Denote MCR_s and PCR_s the minimum rate requirement and the peak rate constraint for each connection $s \in \mathcal{S}$, respectively. We say that a rate vector $r = \{r_s \mid s \in \mathcal{S}\}$ is *ABR-feasible* if the following two constraints are satisfied:

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

For feasibility, we assume that the sum of all connections' MCR requirements traversing any link does not exceed the link's capacity, i.e.

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

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

In our generic weight-based max-min policy, we associate each connection $s \in \mathcal{S}$ with a weight (or priority) w_s.[1] Informally, the WMM policy first allocates to each connection its MCR. Then from the remaining network capacity, it allocates additional bandwidth for each connection using a proportional version of the max-min policy based on each connection's weight while satisfying its PCR constraint. The final bandwidth for each connection is its MCR plus an additional "weighted" max-min share. The following is a centralized algorithm for this policy.

Algorithm 1 A Centralized Algorithm

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

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

Formally, this policy is characterized by the following two definitions and two theorems.

Definition 1 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 connection $t \in \mathcal{S}$ such that $\frac{r_s - \text{MCR}_s}{w_s} \ge \frac{r_t - \text{MCR}_t}{w_t}$ and $r_t > \hat{r}_t$. □

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 connection 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 connections t traversing link ℓ. □

Theorem 1 An ABR-feasible rate vector r is WMM if and only if each connection 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. □

Due to paper the length limitation, we refer interested readers to [7] for the proofs of Theorems 1 and 2, as well as a correctness proof of Algorithm 1.

We reiterate that the weight associated with each connection is generic under our WMM policy. The MCRadd [6, 9, 16] and the MCRprop [8, 9, 16] policies are special cases of our WMM policy since MCRadd treats all connections with equal weight while MCRprop assigns each connection's weight the same as its MCR.

The following simple example illustrates how Algorithm 1 allocates network bandwidth for the WMM policy.

Example 1 A Peer-to-Peer Network

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

Table 2 shows the iterations of using Algorithm 1 to allocate a rate for each connection under the WMM policy. □

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

Connection	MCR	PCR	Weight	WMM Rate Allocation
VC1	0.15	1.00	3	0.525
VC2	0.10	0.30	2	0.300
VC3	0.05	0.50	1	0.175

Table 2: Iterations of using the centralized algorithm to calculate WMM rate allocation for each connection in the peer-to-peer network.

Iterations	VC{(MCR, PCR), W}			Remaining Capacity
	VC1{(0.15, 1.00), 3}	VC2{(0.10, 0.30), 2}	VC3{(0.05, 0.50), 1}	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

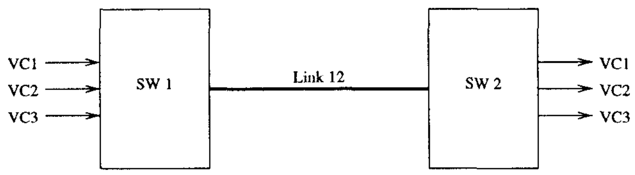

Figure 1: A peer-to-peer network.

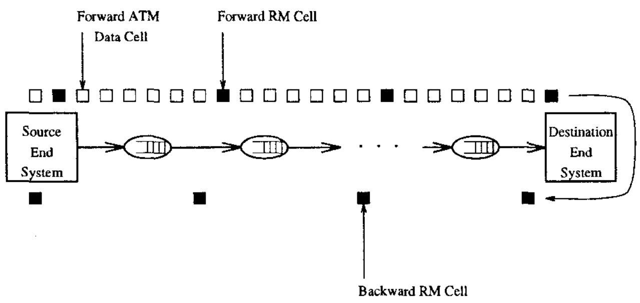

Figure 2: ABR flow control for a virtual connection.

We would like to point out that our WMM policy provides an attractive pricing strategy for network service providers. In particular, each connection may be charged a premium rate corresponding to the guaranteed bandwidth (i.e. MCR). Beyond this rate, each connection may be billed an additional tariff for the weight (or priority) to share any additional unguaranteed (or available) network capacity.

The centralized algorithm for the WMM rate allocation requires global information. It is intended to be used as the network bandwidth sharing optimality criterion for our distributed ABR algorithm, which will be presented in the next section.

3 A Distributed ABR Algorithm

Our objective in this section is to design an ABR algorithm with the aim of converging to the WMM policy through distributed and asynchronous iterations.

A generic ABR flow control mechanism for a virtual connection is shown in Fig. 2. Resource Management (RM) cells are inserted among data cells to convey information between the sources and the network. In the forward path, a source sets the fields in the forward RM cells to inform the network about the source's rate information (e.g. MCR, PCR, CCR). In the backward path, the network switches set the fields (e.g. ER) in the returning RM cells to inform the source about available bandwidth. Upon receiving a backward RM cell, the source adapts its transmission rate to the feedback rate.

Among the many prior efforts on the design of ABR algorithms to achieve the classical max-min [5, 11, 12, 13, 14, 15], the work by Charny *et al.* [5] was one of the few algorithms that were proven to converge to the max-min. We will extend Charny's *Consistent Marking* technique to design an ABR algorithm for our WMM policy, which supports the minimum rate, peak rate, and weight for each connection.

We first specify the end system behavior of our ABR algorithm, which conforms to the ABR framework in [2].

Algorithm 2 End System Behavior

Source Behavior[2]

- 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, W)

[2] We use a simplified version of source and destination behavior, which does not include the use-it-or-lose-it option [2]. We use some unspecified field in the RM cell to carry the weight (W) of the connection.

cell with: CCR := ACR; MCR := MCR; ER := PCR; W := W;

- Upon the receipt of a backward RM(CCR, MCR, ER, W) cell from the destination, the ACR at the source is adjusted to: ACR := ER.

Destination Behavior

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

For the design of our switch algorithm, we employ per-VC accounting at each output port of a switch. That is, we maintain a table at each output port of a switch to keep track of the state information of each traversing connection. Based on the state information of each connection, we calculate the explicit rate for each connection to achieve the WMM rate allocation.

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 connections traversing link ℓ, $\ell \in \mathcal{L}$.
n_ℓ: Number of connections in $\mathcal{G}_\ell$, $\ell \in \mathcal{L}$, i.e., $n_\ell = |\mathcal{G}_\ell|$.
r_ℓ^i: CCR value of connection $i \in \mathcal{G}_\ell$ at link ℓ.
MCR^i: MCR requirement of connection i.
b_ℓ^i: Bit used to mark connection $i \in \mathcal{G}_\ell$ at link ℓ.

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

$\mathcal{Y}_\ell$: Set of connections 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 connections 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: Variable used to estimate the WMM-bottleneck link rate.

The following is our switch algorithm, with each output port link initialized with: $\mathcal{G}_\ell = \emptyset$; $n_\ell = 0$; and $\mu_\ell = \infty$.

Algorithm 3 Switch Behavior

Upon the receipt of a forward RM(CCR, MCR, ER, W) cell from the source of connection i {
  if RM cell signals connection termination[3]{
    $\mathcal{G}_\ell := \mathcal{G}_\ell - \{i\}$; $n_\ell := n_\ell - 1$;
    table_update();
    }
  if RM cell signals connection initiation {
    $\mathcal{G}_\ell := \mathcal{G}_\ell \cup \{i\}$; $n_\ell := n_\ell + 1$;
    $r_\ell^i :=$ CCR; $\text{MCR}^i :=$ MCR; $w_i :=$ W;
    $b_\ell^i := 0$;
    table_update();
    }
  else {
  /* i.e. RM cell belongs to an active connection. */
    $r_\ell^i :=$ CCR;
    if $(\frac{r_\ell^i - \text{MCR}^i}{w_i} \le \mu_\ell)$ then $b_\ell^i := 1$;
    table_update();
    }

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

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

table_update()
{
  rate_calculation_1: use Algorithm 4 to calculate μ_ℓ^1;

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

  /* Update μ_ℓ after the above unmarking operation.*/
  rate_calculation_2: use Algorithm 4 to calculate μ_ℓ;

  if $(\mu_\ell < \mu_\ell^1)$, then {
    Unmark any marked connection $i \in \mathcal{G}_\ell$ at link ℓ with $\frac{r_\ell^i - \text{MCR}^i}{w_i} > \mu_\ell$;
    rate_calculation_3: use Algorithm 4 to calculate μ_ℓ again;
    }[4]
} □

Algorithm 4 μ_ℓ Calculation

$$\mu_\ell := \begin{cases} \infty & \text{if } n_\ell = 0;^5 \\[2ex] \frac{C_\ell - \sum_{i\in\mathcal{G}_\ell} r_\ell^i}{\sum_{i\in\mathcal{G}_\ell} w_i} + \max_{i\in\mathcal{G}_\ell} \frac{r_\ell^i - \text{MCR}^i}{w_i} & \text{if } n_\ell = |\mathcal{Y}_\ell|; \\[2ex] \frac{(C_\ell - \sum_{i\in\mathcal{G}_\ell} \text{MCR}^i) - \sum_{i\in\mathcal{Y}_\ell}(r_\ell^i - \text{MCR}^i)}{\sum_{i\in\mathcal{U}_\ell} w_i} & \text{otherwise.} \end{cases}$$

□

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.

[3] This information is conveyed through some unspecified bits in the RM cell, which can be set either at the source or the UNI.
[4] Both μ_ℓ^1 and μ_ℓ follow the same μ_ℓ calculation in Algorithm 4.
[5] In fact, μ_ℓ can be set to any value when $n_\ell = 0$.

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 $\text{MCR}^s \leq \text{CCR}^s \leq \text{PCR}^s$. □

The key concept used in the convergence proof of our distributed ABR algorithm is the notion of *marking consistent*, which is defined as follows.

Definition 3 Let $\mathcal{Y}_\ell$ be the set of connections that are marked at link $\ell \in \mathcal{L}$ and μ_ℓ be calculated according to Algorithm 4. The marking of connections at link $\ell \in \mathcal{L}$ is *marking-consistent* if

$$\frac{r_\ell^i - \text{MCR}^i}{w_i} \leq \mu_\ell$$

for every connection $i \in \mathcal{Y}_\ell$. □

It can be shown that by using the three-step rate calculation for μ_ℓ in the "table_update()" subroutine of Algorithm 3, the marking of all connections at a link satisfies the marking-consistent property after the switch algorithm is performed for each RM cell traversing this link [7].

Denote M the total number of iterations needed to execute Algorithm 1. It can be shown that $M \leq |\mathcal{S}|$, where $|\mathcal{S}|$ is the total number of connections in the network [7]. Let $\mathcal{S}_i$, $1 \leq i \leq M$ be the set of connections being removed at the end of the ith iteration, i.e. connections in $\mathcal{S}_i$ have either reached their WMM-bottleneck link rate or their PCRs during the ith iteration of Algorithm 1. Let τ_i, $1 \leq i \leq M$ be defined as follows:

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

where r^s is the final WMM rate allocation for connection s by Algorithm 1. By the operation of Algorithm 1, for a connection $p \in \mathcal{S}$ which has not yet gone through a saturated link or reached its PCR, its $\frac{r^p - \text{MCR}^p}{w_p}$ increases at each iteration. Therefore, we have $\tau_1 < \tau_2 < \ldots < \tau_M$.

It can be shown that after some finite time T_1, the set of connections in $s \in \mathcal{S}_1$ will either reach their WMM-bottleneck link rate or their PCR constraints. These connections will be allocated with their optimal rates permanently and are marked at every link they traverse. By the operation of our rate calculation in the switch algorithm, such marked connections (as well as their associated bandwidth) can be used as the base case of an induction argument for the convergence of the second level WMM rate allocation (i.e. $s \in \mathcal{S}_2$). Using the same taken (i.e. induction), it can be shown that eventually all connections in the network will reach their WMM rate allocation and will be marked at every link they traverses.

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

Corollary 3.1 Let D be the maximum round-trip time among all connections. Then an upper bound on the convergence time to the WMM policy by our ABR algorithm from the time when the number of active connections in the network stabilizes is given by $2.5MD$. □

Due to paper length limitation, we refer interested readers to [7] for a complete formal proof of Theorem 3 and Corollary 3.1.

4 Simulation Results

Theorem 3 provides for the guaranteed convergence to the WMM rate allocation for our distributed ABR algorithm under any network configuration and any set of link distances. In this section, we perform simulations on various benchmark network configurations suggested by the ATM Forum Traffic Management Group to demonstrate the fast convergence property of our ABR algorithm.

The ATM switches in the simulations are assumed to have output port buffering with internal switching capacity equal to the aggregate rates of its input ports. Each output port employs the simple FIFO queuing discipline and is shared by all VCs going through that port. We set the link capacity to be 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. By setting a target link utilization strictly less than 1, we ensure that the potential buffer build up during transient period will be emptied upon convergence. The distance from an end system (source or 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. At each source, we let Nrm = 32 and ICR := MCR.

The Peer-to-Peer Network

For this network (Fig. 1), the output port link of SW1 is the only bottleneck link for the three connections.

Under a normalized unit link capacity, the minimum rate requirement, peak rate constraint, weight, and WMM rate allocation of each connection are listed in Table 1.

Fig. 4 shows the ACR at source for connections VC1, VC2, and VC3, respectively. The cell rates shown in the plot are normalized with respect to the capacity C_ℓ

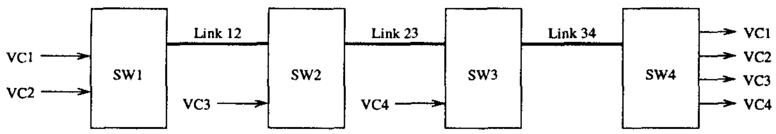

Figure 3: A parking lot network.

Table 3: MCR requirement, PCR constraint, weight, and WMM rate allocation for each connection in the parking lot network.

Connection	MCR	PCR	Weight	WMM Rate Allocation
VC1	0.15	0.35	4	0.2543
VC2	0.10	0.20	2	0.1522
VC3	0.10	0.50	8	0.3087
VC4	0.05	0.50	9	0.2848

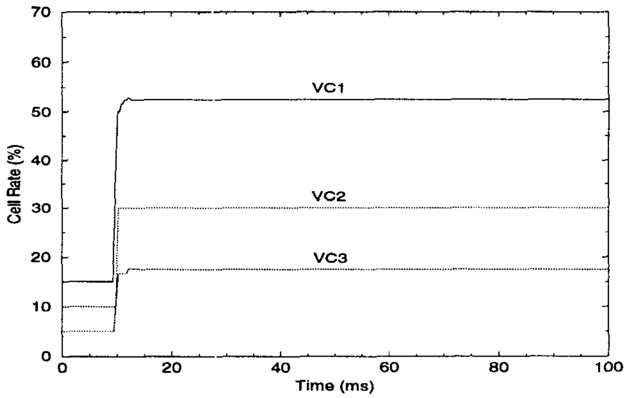

Figure 4: The cell rates of all connections for the peer-to-peer network.

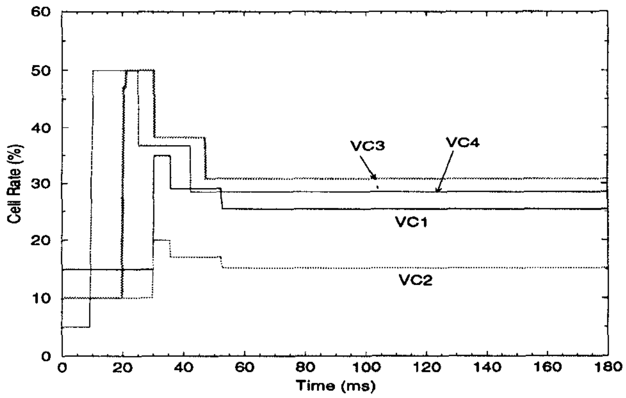

Figure 5: The cell rates of all connections for the parking lot network.

(142.5 Mbps) for easy comparison with those values obtained with our centralized algorithm under unit link capacity in Table 1. Each connection starts with its MCR. The first RM cell for each connection returns to the source after one round trip time (RTT), or 10 ms. After initial iterations, we see that the cell rate of each connection converges to its optimal WMM rate listed in Table 1. Also, we find that during the course of distributed iterations, the ACR of each connection maintains ABR-feasibility, i.e., MCR $\leq$ ACR $\leq$ PCR.

Also shown in Fig 4 is that the convergence time of our ABR algorithm is much faster than the upper bound given in Corollary 3.1. Here the RTT is 10 ms and it takes less than 15 ms for our ABR algorithm to converge.

A Parking Lot Network

The specific parking lot network that we use is shown in Fig. 3, where connections VC1 and VC2 start from the first switch and go to the last switch, and connections VC3 and VC4 start from SW2 and SW3, respectively, and terminate at the last switch [10].

Table 3 lists the MCR requirement, PCR constraint, weight, and WMM rate allocation for each connection in the parking lot network under unit link capacity.

Fig. 5 shows the normalized ACR of each connection under our distributed ABR algorithm. We find that the ACR of each connection converges to its optimal WMM rate listed in Table 3. Here the maximum RTT among all connections is 30 ms (VC1 and VC2) and it takes our distributed ABR algorithm less than 2 RTT to converge to the final optimal rates.

A Generic Fairness Network

The specific generic fairness configuration that we use is shown in Fig. 6 where there are five ATM switches connected in a chain with six paths traversing these ATM switches and sharing link capacity [4].

Table 4 lists the MCR requirement, PCR constraint, weight, and WMM rate allocation for each connection under unit link capacity.

Fig. 7 shows the normalized cell rate of each connection under our distributed ABR algorithm. Again, the

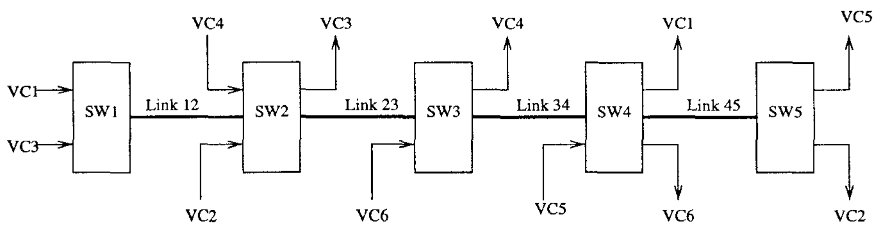

Figure 6: A generic fairness network.

Table 4: MCR requirement, PCR constraint, weight, and WMM rate allocation for each connection in the generic fairness network.

Connection	MCR	PCR	Weight	WMM Rate Allocation
VC1	0.10	1.00	4.5	0.3077
VC2	0.20	1.00	4.0	0.3846
VC3	0.20	0.60	2.0	0.6000
VC4	0.05	0.55	2.5	0.3077
VC5	0.05	0.85	4.0	0.6154
VC6	0.10	1.00	4.5	0.3077

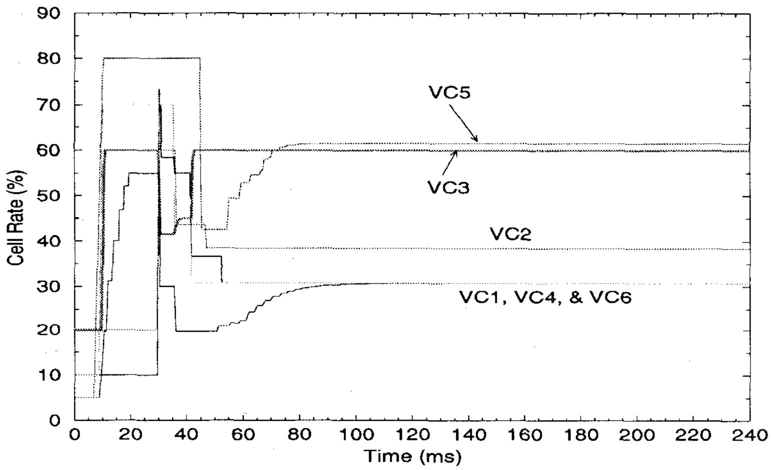

Figure 7: The cell rates of all connections for the generic fairness network.

rate of each connection converges to its optimal WMM rate listed in Table 4. Here the maximum RTT among all connections is 30 ms (VC1 and VC2) and it takes less than 4 RTT for our ABR algorithm to converge.

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

5 Concluding Remarks

We presented a novel generic weight-based rate allocation policy to share network bandwidth among ABR connections. Our policy supports a minimum rate for each connection and shares any remaining network bandwidth among all connections based on a weight associated with each connection, while satisfying its peak rate constraint. Our WMM rate allocation policy also offers an attractive pricing model to the network service providers wishing to introduce priority options to user connections in a usage-based pricing policy.

We designed a distributed ABR algorithm to achieve our WMM rate allocation. Our ABR algorithm was proven to provide guaranteed convergence to our rate allocation policy under any network configuration and any set of link distances through distributed and asynchronous iterations. Simulation results demonstrated the fast convergence property of our ABR algorithm.

Our future work will focus on other issues of our ABR algorithm 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 give a rigorous proof of the algorithm's convergence. Other issues include system transient behavior, rate of convergence, and network buffer requirements, which should all be carefully investigated before deploying an ABR algorithm for ATM networks.

References

[1] A. Arulambalam, X. Chen, and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM Networks," *IEEE Communications Magazine*, pp. 92–100, November 1996.

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

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

[4] F. Bonomi and K. W. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service," *IEEE Network*, vol. 9, no. 2, pp. 25–39, March/April 1995.

[5] A. Charny, D. Clark, and R. Jain, "Congestion Control with Explicit Rate Indication," *Proc. IEEE ICC'95*, pp. 1954–1963, Seattle, WA, June 18–22, 1995.

[6] 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.

[7] 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.

[8] Y. T. Hou, H. Tzeng, and S. S. Panwar, "Weighted Max-Min Fairness for ABR Service: Theory and Implementation," *Proc. IEEE Global Communications Conference (GLOBECOM'97)*, pp. 492–497, Phoenix, AZ, USA, Nov. 4–8, 1997.

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

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

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

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.