Generalized Max-Min Rate Allocation: Theory and a Simple Implementation

Yiwei Thomas Hou, Shivendra S. Panwar, and Henry Tzeng

Abstract: **An important concept in the available bit rate (ABR) service model is the minimum cell rate (MCR) guarantee as well as the peak cell rate (PCR) constraint for each flow. Due to the MCR and PCR requirements, the well-known max-min rate allocation policy no longer suffices to determine the rate allocation for each flow since it does not support either MCR or PCR. In this paper, we present a generalized max-min (GMM) rate allocation policy, which supports both the MCR and PCR requirements for each flow. Furthermore, a simple distributed algorithm using the ABR flow control protocol is developed to achieve the GMM rate allocation in a distributed network environment. The effectiveness of this distributed algorithm is demonstrated by simulation results.**

Index Terms: **Max-min rate allocation, available bit rate, minimum rate, peak rate, flow control.**

I. INTRODUCTION

One key performance objective for flow-oriented packet-switched networks is to optimally share network bandwidth among all traffic flows. Optimality here typically includes two components: 1) each flow is entitled to as much network bandwidth as any other flow (i.e., fairness), and 2) the network bandwidth is utilized efficiently.

The classical max-min rate allocation has been widely regarded as an optimal bandwidth sharing policy among traffic flows in the network [1]. In ATM networks, the max-min rate allocation has been used for the available bit rate (ABR) service [2]. By the specifications in [2], on the establishment of an ABR flow, the user shall specify to the network both a minimum rate and a maximum rate, designated as minimum cell rate (MCR) and peak cell rate (PCR), respectively, for the requested flow. The source starts to transmit at an initial cell rate (ICR), which is greater than or equal to MCR, and may increase its rate up to PCR depending upon congestion and bandwidth information from the network.

Since the classical max-min does not address how to determine rate allocation for each flow when there are MCR and PCR constraints, an MCR-offsetted and an MCR-weighted max-min policies were proposed in [3], [4].

In this paper, we generalize the classical max-min rate allocation. We present a generalized max-min (GMM) rate allocation policy with both the minimum rate guarantee and the peak rate constraint for each flow. We also present a centralized bandwidth assignment algorithm to achieve the GMM policy and prove its correctness.

Based on the centralized theory for the GMM policy, we develop a distributed algorithm to achieve GMM rate allocation. Our distributed algorithm employs the ABR flow control protocol and is based on the *Intelligent Marking* technique by Siu and Tzeng [5], [6] which was developed for the classical max-min. We use simulation results to demonstrate the effectiveness of our distributed algorithm to achieve GMM rate allocation.

The remainder of this paper is organized as follows. In Section II, we present the theory of GMM rate allocation to generalize the classical max-min with both the minimum rate guarantee and the peak rate constraint for each flow. In Section III, we develop a distributed algorithm using ABR flow control protocol to achieve the GMM policy. In Section IV, we present simulation results to demonstrate the effectiveness of the distributed algorithm. Section V concludes this paper.

II. GENERALIZED MAX-MIN RATE ALLOCATION

We organize this section as follows. In Section II-A, we briefly summarize key results for the classical max-min rate allocation [1]. In Section II-B, we present the GMM rate allocation, which generalizes the classical max-min. A centralized algorithm to determine rate allocation for the GMM policy is then presented in Section II-C.

A. Preliminaries

In our model, a network $\mathcal{N}$ is characterized by a set of links $\mathcal{L}$ and flows $\mathcal{S}$. Each flow $s \in \mathcal{S}$ traverses one or more links in $\mathcal{L}$ and is allocated a specific rate r^s. Denote $\mathcal{S}_\ell$ the set of flows traversing link ℓ. Then the (aggregate) allocated rate F_ℓ on link $\ell \in \mathcal{L}$ of the network is $F_\ell = \sum_{s \in \mathcal{S}_\ell} r^s$.

Let C_ℓ be the capacity of link ℓ. A link ℓ is *saturated* or *fully utilized* if $F_\ell = C_\ell$. A rate vector $r = \{r^s \mid s \in \mathcal{S}\}$ is *feasible* if the following two constraints are satisfied: 1) $r^s \geq 0$ for all $s \in \mathcal{S}$; and 2) $F_\ell \leq C_\ell$ for all $\ell \in \mathcal{L}$.

A rate vector r is *max-min* if it is feasible, and for each flow s, one cannot generate a new feasible rate vector by increasing the allocated rate r^s without decreasing the allocated rate of some other flow t with a rate r^t already less than or equal to r^s in the rate vector r. More formally, we have the following.

Manuscript received November 26, 1999; approved for publication by Bo Li, Division III Editor, April 26, 2000.

Y. T. Hou is with Fujitsu Laboratories of America, Sunnyvale, CA, USA, e-mail: thou@fla.fujitsu.com. This work was completed while Y. T. Hou was with Polytechnic University, Department of Electrical Engineering, Brooklyn, NY, USA.

S. S. Panwar is with Polytechnic University, Department of Electrical Engineering, Brooklyn, NY, USA.

H. Tzeng is with Amber Networks Inc., Santa Clara, CA, USA.

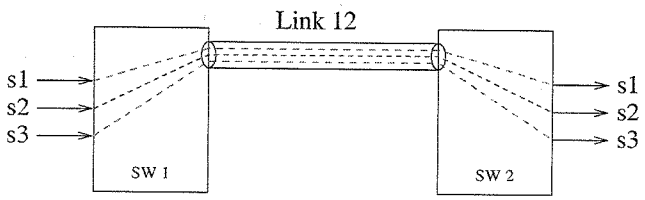

Fig. 1. A peer-to-peer network.

Definition 1: A rate vector r is max-min if it is feasible, and for each $s \in \mathcal{S}$ and every feasible rate vector $\hat{r}$ in which $\hat{r}^s > r^s$, there exists some flow $t \in \mathcal{S}$ such that $r^s \geq r^t$ and $r^t > \hat{r}^t$.

Definition 2: Given a feasible rate vector r, a link $\ell \in \mathcal{L}$ is a bottleneck link with respect to r for a flow s traversing ℓ if $F_\ell = C_\ell$ and $r^s \geq r^t$ for all flows t traversing link ℓ.

Theorem 1: A feasible rate vector r is max-min if and only if each flow has a bottleneck link with respect to r.[1]

Algorithm 1: The following iterative steps describes the centralized algorithm to determine rate allocation for each flow for the classical max-min policy.

1. Start the rate allocation of each flow with zero.
2. Increase the rate of all flows with the smallest rate such that some link becomes saturated.
3. Remove those flows that traverse saturated links and their associated bandwidth from the network.
4. If there is no flow left, the algorithm terminates; otherwise, go back to Step 2 for the remaining flows and remaining network capacity.

Theorem 2: There exists a unique rate vector that satisfies the max-min rate allocation.

A proof of Theorem 2 is given in the Appendix.

B. Generalized Max-Min Rate Allocation

We are now ready to formally define the generalized max-min rate allocation policy with minimum rate and peak rate constraints for each flow.

Let MCR^s and PCR^s be the minimum rate requirement and the peak rate constraint for each flow $s \in \mathcal{S}$. For the sake of feasibility, we make the following assumption.

Assumption 1: The sum of all the flows' MCR requirements traversing any link does not exceed the link's capacity, i.e., $\sum_{s \in \mathcal{S}_\ell} \text{MCR}^s \leq C_\ell$ 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 flow.

Definition 3: A rate vector $r = \{r^s | s \in \mathcal{S}\}$ is ABR-feasible if the following two constraints are satisfied:

$$\text{MCR}^s \leq r^s \leq \text{PCR}^s, \qquad \text{for all } s \in \mathcal{S},$$
$$F_\ell \leq C_\ell, \qquad \text{for all } \ell \in \mathcal{L}.$$

Before we give a definition for the GMM rate allocation, we use the following simple example to illustrate its basic concept.

In the *peer-to-peer* network configuration (Fig. 1), the output port link of SW1 (Link12) is the only bottleneck link for all flows. Assume that all links have the same normalized one unit of capacity. When there is no MCR and PCR requirements for each flow, using the centralized max-min algorithm described in Algorithm 1, we allocate to each flow with rate of $\frac{1}{3}$. Now let the MCR requirement and PCR constraint for each flow be as listed in Table 1. It is clear that the classical max-min is no longer applicable here because it does not support either the MCR or the PCR.

Table 1. MCR, PCR, and GMM rate allocation of each flow in the peer-to-peer network.

Flow	MCR	PCR	GMM Rate Allocation
$s1$	0.40	1.00	0.40
$s2$	0.10	0.25	0.25
$s3$	0.05	0.50	0.35

In the following, we describe the iterative steps of the centralized algorithm to determine rate allocation for each flow under the generalized max-min policy, which we will formally define (Definition 4) shortly.

Algorithm 2: This algorithm describes how to determine the rate allocation for each flow under GMM.[2]

1. Start the rate of each flow with its MCR.
2. Identify the flow with the smallest rate among all the remaining flows and increase the rate of such flow(s) until one of the following events first takes place:
   - the rate of such flow(s) reaches the second smallest rate among all the remaining flows;
   - some link saturates;
   - the rate of such flow(s) reaches its PCR.
3. If some link saturates or such flow reaches its PCR in Step 3, remove such flows that either traverse this saturated link or reach their PCRs, respectively, as well as the network capacity associated with such flows from the network.
4. If there is no flow left, the algorithm terminates; otherwise, go back to Step 2 for the remaining flows and network capacity.

With the above centralized algorithm for the GMM rate allocation. we are able to complete the rate allocation problem for the peer-to-peer network configuration (Fig. 1) with the MCR and PCR requirements listed in Table 1.

Example 1: (A Peer-to-Peer Network) Using Algorithm 2, we list the rate allocation for each flow at each iteration in Table 2, which are explained briefly as follows. A graphical display of the iterative steps in Table 2 is also shown in Fig. 2.

Step 1 – As shown in Fig. 2 and Table 2, we start the rate allocation for each flow with its MCR requirement (shown in the darkest shaded areas in Fig. 2).

Step 2 – Since the rate of $s3$ (0.05) is the smallest among all flows, we increase it until it reaches the second smallest rate, which is 0.1 ($s2$).

Step 3 – The rates of both $s2$ and $s3$ being 0.1, we increase them together until $s2$ reaches its PCR constraint of 0.25.

[1] For a proof of Theorem 1, see [1].

[2] A formal mathematical description of this algorithm will be given in Section II-C.

Table 2. Iterations of centralized rate allocation algorithm for the peer-to-peer network configuration.

Iterations	Flow(MCR, PCR)			Remaining Link Capacity
	$s1$(0.40, 1.00)	$s2$(0.10, 0.25)	$s3$(0.05, 0.50)	Link 12
Initialization	0.40	0.10	0.05	0.45
1st	0.40	0.10	0.10	0.40
2nd	0.40	0.25	0.25	0.10
3rd	0.40		0.35	0

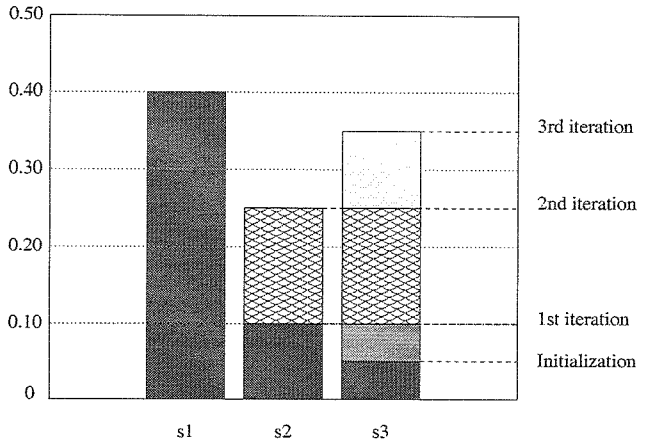

Fig. 2. Graphical display of rate allocation for each flow at each iteration in the peer-to-peer example.

Step 4 – Remove $s2$ (with a rate of 0.25) out of future iterations and we now have the rates of 0.40 and 0.25 for $s1$ and $s3$, respectively, with a remaining capacity of 0.10 on Link 12.

Step 5 – Since $s3$ has a smaller rate (0.25) than $s1$ (0.4), we increase the rate of $s3$ to 0.35 and Link 12 saturates. The final rate assignments are 0.40, 0.25, and 0.35 for $s1$, $s2$, and $s3$, respectively.

Informally, a rate vector r is GMM if it is ABR-feasible, and for each flow s, one cannot generate a new ABR-feasible rate vector $\hat{r}$ by increasing the allocated rate r^s without decreasing the allocated rate of some other flow t with a rate r^t already less than or equal to r^s in the rate vector r. Formally, the GMM rate allocation is defined as follows.

Definition 4: A rate vector r is Generalized Max-Min (GMM) if it is ABR-feasible, and for every $s \in \mathcal{S}$ and every ABR-feasible rate vector $\hat{r}$ in which $\hat{r}^s > r^s$, there exists some flow $t \in \mathcal{S}$ such that $r^s \geq r^t$, and $r^t > \hat{r}^t$.

Since there are MCR and PCR requirements for each flow, we define a new notion of bottleneck link as follows.

Definition 5: Given an ABR-feasible rate vector r, a link $\ell \in \mathcal{L}$ is an GMM-bottleneck link with respect to r for a flow s traversing ℓ if $F_\ell = C_\ell$ and $r^s \geq r^t$ for every flow t traversing link ℓ for which $r^t > \text{MCR}^t$.

Theorem 3: An ABR-feasible rate vector r is GMM if and only if each flow has either an GMM-bottleneck link with respect to r or a rate assignment equal to its PCR.

For a proof of Theorem 3, see the Appendix.

In Example 1, Link 12 is an GMM-bottleneck link for both $s1$ and $s3$ (see Definition 5). On the other hand, $s1$ and $s3$ have different rate allocations (0.4 for $s1$ and 0.35 for $s3$). Thus, it is necessary to have a more precise definition for *GMM-bottleneck link rate,* which is presented as follows.

Let $\mathbf{1}_{\{\text{event A}\}}$ be the indicator function with the following definition.

$$\mathbf{1}_{\{\text{event A}\}} = \begin{cases} 1, & \text{if event A is true;} \\ 0, & \text{otherwise.} \end{cases}$$

Definition 6: Given an GMM rate vector r, suppose that link $\ell \in \mathcal{L}$ is an GMM-bottleneck link with respect to r and let τ_ℓ denote the GMM-bottleneck link rate at ℓ. Then τ_ℓ satisfies

$$\tau_\ell \cdot \sum_{i \in \mathcal{U}_\ell} \mathbf{1}_{\{\text{MCR}^i \leq \tau_\ell\}} + \sum_{i \in \mathcal{U}_\ell} \text{MCR}^i \cdot \mathbf{1}_{\{\text{MCR}^i > \tau_\ell\}} = C_\ell - \sum_{i \in \mathcal{Y}_\ell} r_\ell^i,$$

where

- $\mathcal{U}_\ell$ denotes the set of flows that are GMM-bottlenecked at link ℓ;
- $\mathcal{Y}_\ell$ denotes the set of flows that are either GMM-bottlenecked at some other link or have rate assignments equal to their PCRs and $r_\ell^i < \tau_\ell$ for $i \in \mathcal{Y}_\ell$.

It is worth pointing out that in the special case when $\text{MCR}^s = 0$ for every $s \in \mathcal{S}$, the GMM-bottleneck link rate τ_ℓ in Definition 6 becomes:

$$\tau_\ell \cdot |\mathcal{U}_\ell| = C_\ell - \sum_{i \in \mathcal{Y}_\ell} r_\ell^i,$$

or

$$\tau_\ell = \frac{C_\ell - \sum_{i \in \mathcal{Y}_\ell} r_\ell^i}{|\mathcal{U}_\ell|},$$

where $|\mathcal{U}_\ell|$ denotes the number of flows in $\mathcal{U}_\ell$. This is exactly the expression for the classical max-min rate allocation at link ℓ.

With Definition 6, we are now ready to go back to Example 1. It is now straight forward to see that the GMM-bottleneck link rate at Link 12 is 0.35.

C. GMM Centralized Algorithm and More Examples

An intuitive description of the centralized algorithm to determine rate allocation for the GMM was given in Algorithm 2. In the following, we present the formal mathematical description of such a centralized algorithm.

Algorithm 3: The following is a formal mathematical description of the centralized algorithm for GMM rate allocation.
Initialization:

Table 3. A second example of GMM rate allocation for the peer-to-peer network.

Flow	MCR	PCR	GMM Rate Allocation
$s1$	0.40	1.00	0.45
$s2$	0.05	0.10	0.10
$s3$	0.05	0.50	0.45

$r^{s,(0)} = \text{MCR}^s$, for every $s \in \mathcal{S}$,
$F_\ell^{(0)} = \sum_{s \in \mathcal{S}_\ell} \text{MCR}^s$, for every $\ell \in \mathcal{L}$,
$k = 1, \mathcal{S}^{(1)} = \mathcal{S}, \mathcal{L}^{(1)} = \mathcal{L}$.

1. Sort all the flows in $\mathcal{S}^{(k)}$ into m sets ($1 \le m \le |\mathcal{S}^{(k)}|$):
$$u_1, u_2, \cdots, u_m,$$
such that 1) each flow in the same set has the same rate; and 2) rate values in these sets are in increasing order, i.e., $r^s < r^t < \cdots < r^p$, where $s \in u_1, t \in u_2, \cdots, p \in u_m$.
2. $n_\ell^{(k)}$:= number of flows $s \in u_1$ traversing link ℓ, for every $\ell \in \mathcal{L}^{(k)}$.
3. $a^{(k)} :=$
$$\begin{cases} \min\{\min\limits_{\ell \text{ traversed by } s \in u_1} \dfrac{(C_\ell - F_\ell^{(k-1)})}{n_l^{(k)}}, \\ \quad (r^{t \in u_2} - r^{s \in u_1}), \ \min_{s \in u_1}(\text{PCR}^s - r^{s,(k-1)})\}, & \text{if } m > 1, \\ \min\{\min\limits_{\ell \text{ traversed by } s \in u_1} \dfrac{(C_\ell - F_\ell^{(k-1)})}{n_l^{(k)}}, \\ \quad \min_{s \in u_1}(\text{PCR}^s - r^{s,(k-1)})\}, & \text{if } m = 1. \end{cases}$$
4. $r^{s,(k)} := \begin{cases} r^{s,(k-1)} + a^{(k)}, & \text{if } s \in u_1; \\ r^{s,(k-1)}, & \text{otherwise.} \end{cases}$
5. $F_\ell^{(k)} := \sum_{s \in \mathcal{S}_\ell} r^{s,(k)}$, for every $\ell \in \mathcal{L}^{(k)}$.
6. $\mathcal{L}^{(k+1)} := \{\ell \mid C_\ell - F_\ell^{(k)} > 0, \ell \in \mathcal{L}^{(k)}\}$.
7. $\mathcal{S}^{(k+1)} := \{s \mid s$ does not traverse any link $\ell \in (\mathcal{L} - \mathcal{L}^{(k+1)})$ and $r^{s,(k)} \neq \text{PCR}^s\}$.
8. $k := k + 1$.
9. If $\mathcal{S}^{(k)}$ is empty, then $r^{(k-1)} = \{r^{s,(k-1)} \mid s \in \mathcal{S}\}$ is the rate vector satisfying the GMM policy and the algorithm terminates; otherwise, go back to Step 3.

The correctness proof of Algorithm 3 is given in the Appendix. Note that in Step 1 of Algorithm 3, if $m > 1$, then the rate values in $u_2, \cdots, u_m$ are the MCR values of the flows in these sets. Also, starting from the second iteration ($k = 2$), the sorting procedure in Step 1 only requires minor updates based on the sorted sets in the previous iteration.

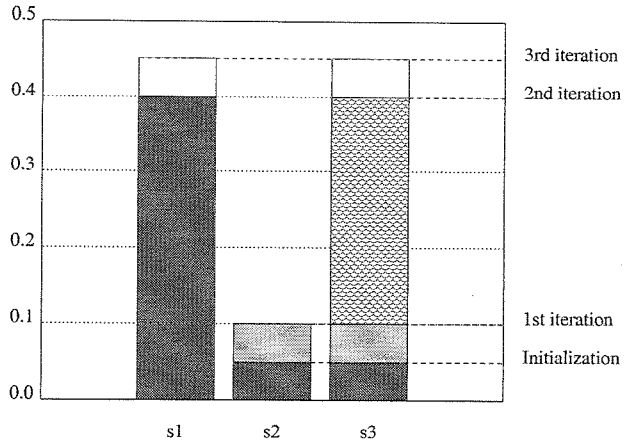

Fig. 3. Graphical display of rate allocation for each flow at each iteration for the second example of the peer-to-peer network.

It is clear that by Definition 6 and Algorithm 3, the GMM rate assignment for a flow $s \in \mathcal{S}$ can only be one of the following:

- a rate assignment equal to its MCR;
- a rate assignment equal to its GMM-bottleneck link rate;
- a rate assignment equal to its PCR.

Using arguments similar to those in the proofs of Theorems 2 and 3, it can be shown that the following theorem is true.

Theorem 4: There exists a unique rate vector that satisfies the GMM rate allocation.

We use the following examples to illustrate how Algorithm 3 allocates network bandwidth such that the GMM rate allocation is satisfied.

Example 2: (A Second Example for the Peer-to-Peer Network) This is the same network configuration (Fig. 1) as for Example 1. The MCR and PCR requirements for each flow are listed in Table 3. The rate allocation for each flow at each iteration is listed in Table 4, with a graphical display in Fig. 3. The GMM-bottleneck link rate at Link 12 is 0.45. Since $s2$ is constrained by its PCR, it is equivalent to treating such a flow as being bottlenecked at some other link, e.g. its access link to the network.

We use the following three-node example to further illustrate the concept of GMM-bottleneck link rate. As we shall see, the GMM-bottleneck link rate of each GMM-bottleneck link being reached during the iterations of Algorithm 3 has the property of ascending order.

Example 3: (A Three-Node Network) In this network configuration (Fig. 4), the output port links of SW1 (Link 12) and

Table 4. Iterations of the centralized GMM rate allocation algorithm for the second example of the peer-to-peer network.

Iterations	Flow(MCR, PCR)			Remaining Link Capacity
	$s1$(0.40, 1.00)	$s2$(0.05, 0.10)	$s3$(0.05, 0.50)	Link 12
Initialization	0.40	0.05	0.05	0.50
1st	0.40	[0.10]	0.10	0.40
2nd	0.40		0.40	0.10
3rd	[0.45]		[0.45]	0

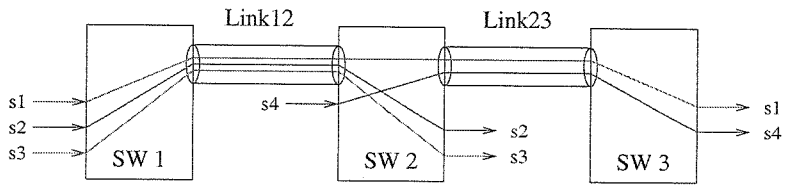

Fig. 4. A three-node network.

Table 5. MCR requirement, PCR constraint, and GMM rate allocation for each flow under the three-node network.

Flow	MCR	PCR	GMM Rate Allocation
$s1$	0.20	0.50	0.425
$s2$	0.05	0.15	0.150
$s3$	0.10	0.50	0.425
$s4$	0.50	1.00	0.575

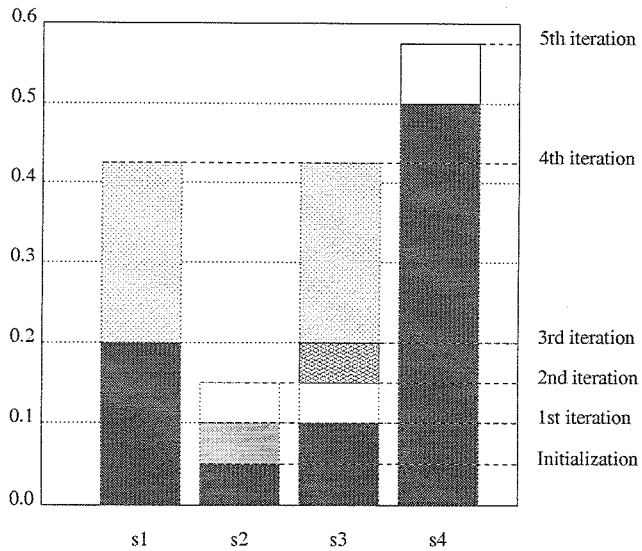

Fig. 5. Graphical display of GMM rate allocation for each flow for the GMM at each iteration in the three-node example.

SW2 (Link 23) are the links shared by flows. The MCR requirement and PCR constraint for each flow are listed in Table 5. The rate allocation for each flow at each iteration of Algorithm 3 is listed in Table 6, with a graphical display in Fig. 5.

The GMM-bottleneck link rate at Link 12 is 0.425, which was reached at the end of the 4th iteration, and the GMM-bottleneck link rate at Link 23 is 0.575, which was reached at the end of the 5th iteration, and $0.425 < 0.575$. In general, by the operation of Algorithm 3, an GMM-bottleneck link rate obtained at a later iteration for some link is greater than an GMM-bottleneck link rate obtained at an earlier iteration for some other link.

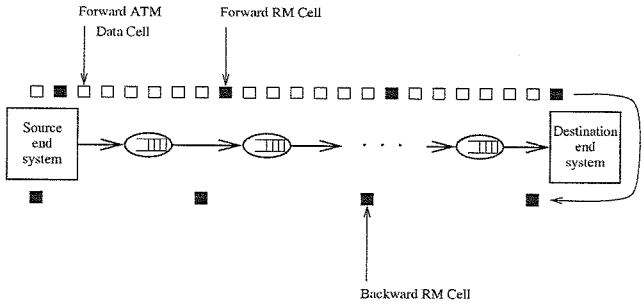

Fig. 6. A schematic for ABR flow control protocol.

Thus far we have completed the centralized theory of the GMM rate allocation. To demonstrate its practical merit for a distributed network, we design a distributed algorithm using the ABR flow control protocol [2] in the next section.

III. A DISTRIBUTED ALGORITHM FOR GMM RATE ALLOCATION

A schematic for ABR flow control protocol is shown in Fig. 6. 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 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. All RM cells of an ABR flow are turned back towards its source after arriving at the destination. A transit node (either along the forward direction or along the backward direction) and destination end system may set the ER field, CI and NI bits in the RM cells.

Table 6. Iterations of using the centralized algorithm for GMM in the the three-node network.

Iterations	Flow(MCR, PCR)				Remaining Link Capacity	
	$s1$ (0.20, 0.50)	$s2$ (0.05, 0.15)	$s3$ (0.10, 0.50)	$s4$ (0.50, 1.00)	Link 12	Link 23
Initialization	0.20	0.05	0.10	0.50	0.65	0.30
1st	0.20	0.10	0.10	0.50	0.60	0.30
2nd	0.20	[0.15]	0.15	0.50	0.50	0.30
3rd	0.20		0.20	0.50	0.45	0.30
4th	[0.425]		[0.425]	0.50	0	0.075
5th				[0.575]		0

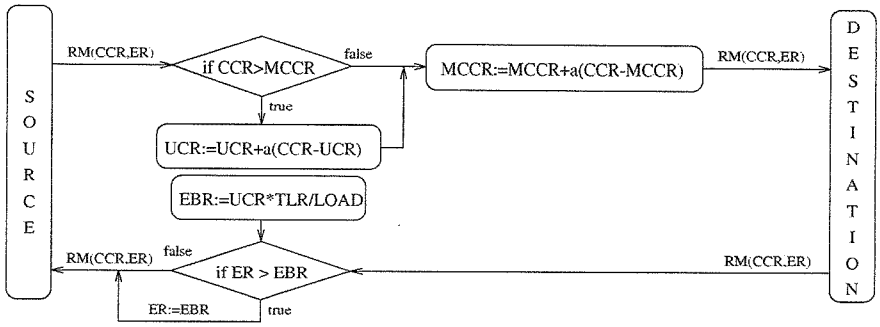

Fig. 7. Switch behavior of Intelligent Marking protocol.

Upon receiving backward RM cells, the source adjusts its cell generating rate accordingly.

There are two modes of switch algorithms, namely, binary mode and explicit rate (ER) mode. Binary schemes (e.g., explicit forward congestion indication (EFCI) [9]) rely on a single bit feedback to indicate congestion. Due to limited feedback information about network congestion status, the source only knows that either the congestion in the network is present or absent, but doesn't know *how much* to increase or decrease its transmission rate. Therefore, the source's cell rate experiences oscillations. On the other hand, ER schemes employ rate calculation at a switch to estimate available bandwidth and convey this information through the ER field in the returning RM cells. Hence, an ER scheme promises higher efficiency and stability than a binary scheme.

Our switch algorithm for the GMM policy employs the ER mode and is based on the *Intelligent Marking* technique, originally proposed in [8] and further refined in [5], [6]. The key idea of this technique is to employ several variables at each link of a switch to estimate the max-min bottleneck link rate with a small number of computations. Using the ABR flow control protocol, 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, resulting in approximate max-min rate allocation.

Fig. 7 illustrates the switch behavior of the Intelligent Marking technique [5], [6]. Four variables, MCCR (Mean Current Cell Rate), UCR (Upper Cell Rate), EBR (Estimated Bottleneck Rate), and LOAD, are defined for the following purpose: 1) MCCR contains an estimated average cell rate of all flows traversing this link; 2) UCR contains an estimated upper limit of the cell rates of all flows traversing this link; 3) EBR contains an estimated bottleneck link rate; 4) LOAD corresponds to the aggregated cell rate entering the queue normalized with respect to the link capacity and is measured over a period of time. Furthermore, two parameters TLR and α are defined for each link, where the value of TLR is the target load ratio, and $0 < \alpha < 1$.

Algorithm 4: (Intelligent Marking)

Upon the receipt of RM(CCR, ER) from the source of a flow
    if (CCR > MCCR), then
        UCR := UCR + α (CCR-UCR);
    MCCR := MCCR + α (CCR-MCCR);
    Forward RM(CCR, ER) to its destination.
Upon the receipt of RM(CCR, ER) from the destination of a flow
    EBR := UCR * TLR / LOAD;
    if (QS > QT), then
        EBR := (QT / QS) * EBR;[3]
    if (ER > EBR), then
        ER := EBR;
    Forward RM(CCR, ER) to its source.

The Intelligent Marking algorithm is a heuristic algorithm. We can only give an intuitive explanation on how it works. The RM cells from all flows participate in exponential averaging for MCCR with MCCR := MCCR + α(CCR − MCCR) while only some flows with greater than MCCR (potentially flows bottlenecked at *this* link) participate in UCR averaging. EBR is used to estimate max-min bottleneck link rate and is based on UCR and LOAD variables. Since 1) there can be only one bottleneck rate at a link and it is greater than or equal to any of the flow's rate traversing this link; and 2) the returning RM cell's ER field is set to the minimum of all the bottleneck link rates along its path, the final rate allocation through Intelligent Marking achieves the max-min rate allocation for each flow (see the "if" part of Theorem 1).

Another interesting fact is that MCCR is larger than the algebraic average of each flow's CCR traversing this link. This is because MCCR is updated more frequently by those flows with relatively larger CCR than those with relatively smaller CCR traversing the same link.

The most attractive feature of the Intelligent Marking technique is its scalability and low implementation cost. It does not require each link of a switch to maintain the state information of each flow and has $O(1)$ storage requirements and computational complexity.

So far we have given a detailed description of the Intelligent Marking technique, which was designed to achieve the classical max-min without MCR/PCR support. Let's see how to extend the Intelligent Marking technique for the GMM rate allocation.

Comparing the definitions for the classical max-min (Definition 1) and the GMM (Definition 4), we observe that they are similar—except the additional requirement under GMM that a rate vector must be ABR-feasible (see Definition 3). This motivates us to take the following steps to design a distributed algorithm for the GMM policy.

1. Continue to use Intelligent Marking (Algorithm 4) as

[3]This step is a finer adjustment of the EBR calculation using buffer occupancy information and is not shown in Fig. 7 due to space limitation in the figure. QS is the Queue Size of the output link and QT is a predefined Queue Threshold.

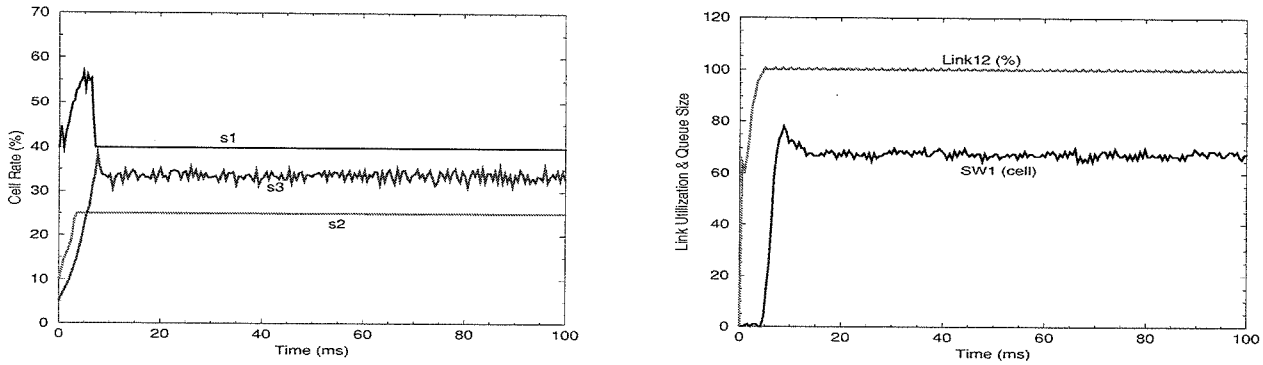

Fig. 8. (a) The cell rates of all flows; (b) the link utilization and queue size of the congested switch for the first example of the peer-to-peer network.

switch algorithm for the GMM policy. This will also satisfy the second requirement for the ABR-feasibility ($F_\ell \leq C_\ell$ for all $\ell \in \mathcal{L}$) due to the nature of Intelligent Marking, i.e., queue size is always kept finite.

2. Let each ABR source enforce the first requirement of ABR-feasibility, i.e., $\text{MCR}^s \leq r^s \leq \text{PCR}^s$ for all $s \in \mathcal{S}$.

The following algorithm specifies the source behavior of our distributed algorithm.

Algorithm 5: (Source Behavior)
The source starts to transmit at ACR := ICR, which is greater than or equal to its MCR; For every N_{rm} transmitted ATM data cells, the source sends a forward RM(CCR, MCR, ER) cell with

CCR := ACR; MCR := MCR; ER := PCR;

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

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

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

IV. SIMULATION RESULTS

In this section, we implement our distributed algorithm on our network simulator and perform simulations to demonstrate its effectiveness in achieving the GMM policy. The switches in all the simulations are assumed to have output buffers with a speedup equal to the number of their ports. The buffer of each output port of a switch employs the simple FIFO queuing discipline and is shared by all flows going through that port.

In all of our simulations, we assume persistent sources. The network configurations that we use are the peer-to-peer and three-node network configurations shown in Figs. 1 and 4, respectively, and the *parking-lot* (Fig. 11) network configuration.

Table 7 lists the parameters used in our simulation. The distance from source/destination to the switch is 100 m and the link distance between switches is 10 km (within the scope of a regional enterprise network).

A. The Peer-to-Peer Network

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

Table 7. Simulation parameters.

End system	PCR	PCR
	MCR	MCR
	ICR	MCR
	Nrm	32
	AIR	3.39 Mbps
Link	Speed	150 Mbps
Switch	Cell switching delay	4 μs
	α	0.125
	Load/utilization measurement interval	500 μs
	Queue threshold for ER adjustment	50 cells
	Output buffer size	2000 cells

Fig. 8(a) shows the ACR at source for flows s1, s2, and s3, respectively with the MCR/PCR requirements for each flow listed in Table 1. The cell rates shown in the plot are normalized with respect to the link capacity (150 Mbps) for easy comparison with those values obtained with our centralized algorithm under unit link capacity (Table 1). After the initial transient period, we see that the cell rate of each flow matches its respective rate in Table 1. To study the network utilization of our distributed algorithm, we also show the inter-switch link utilization (Link 12) and the queue size of congested switch (SW1) in Fig. 8(b). We find that the link is 100% utilized with reasonably small buffer requirements.

Similarly, Fig. 9(a) shows the ACR at source for each flow with the MCR/PCR requirements listed in Table 3 and Fig. 9(b) shows the link utilization and queue size of Link 12. Again, we find that the cell rates under our distributed algorithm match with Table 3 with 100% link utilization and small buffer requirements.

B. The Three-Node Network

For this configuration (Fig. 4), the output port links of SW1 and SW2 are the GMM-bottleneck links.

Fig. 10(a) shows the normalized cell rate of each flow under our distributed algorithm. Comparing with the rates obtained by our centralized algorithm in Table 5, we find that after the initial

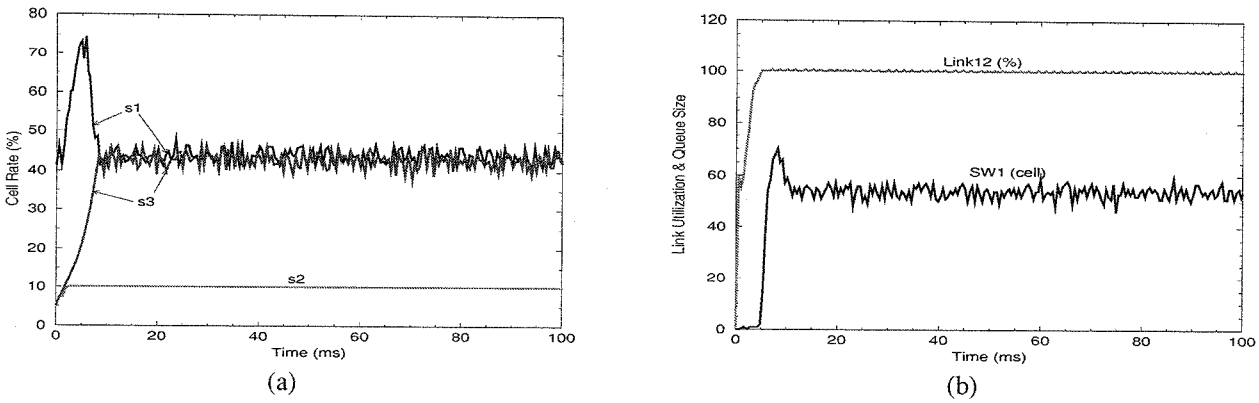

Fig. 9. (a) The cell rates of all flows; (b) link utilization and queue size of the congested switch for the second example of the peer-to-peer network.

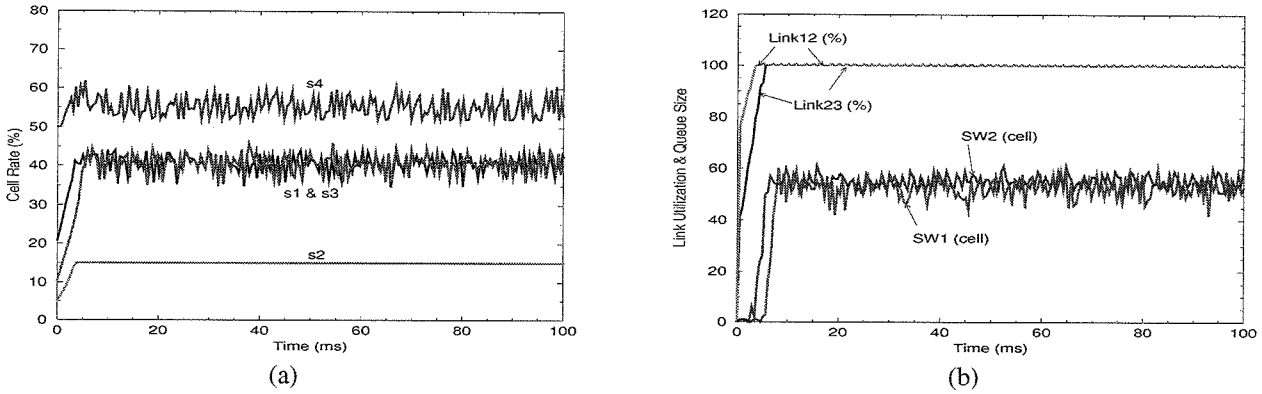

Fig. 10. (a) The cell rates of all flows; (b) the link utilization and queue size at the output ports of congested switches for the three-node network.

Table 8. MCR requirement, PCR constraint, and GMM rate allocation of each flow for the parking-lot network configuration.

Flow	MCR	PCR	GMM Rate Allocation
$s1$	0.05	0.50	0.225
$s2$	0.05	0.15	0.150
$s3$	0.10	0.50	0.225
$s4$	0.40	0.50	0.400

transient period, the rate allocation through our distributed algorithm matches quite well with the GMM rate allocation listed in Table 5.

Fig. 10(b) shows the link utilization at Link 12 and Link 23 as well as the buffer occupancy at output ports of SW1 and SW2. We find that the GMM-bottleneck links are 100% utilized with small buffer requirements.

C. The Parking-Lot Network

The parking-lot configuration that we use is shown in Fig. 11 where flows $s1$ and $s2$ start from the first switch and go to the last switch. $s3$ and $s4$ start from SW2 and SW3, respectively, and terminate at the last switch. Here, Link 34 is the only GMM-bottleneck link.

Table 8 lists the MCR and PCR constraints for each flow and the rate assignment for each flow under the centralized GMM rate allocation algorithm. Note that $s2$ is constrained by its PCR, while Link 34 is the GMM-bottleneck link for $s1$, $s3$, and $s4$. The GMM-bottleneck link rate at Link 34 is 0.225.

Fig. 12(a) shows the normalized cell rates of each flow under our distributed algorithm. We see that they match quite well with the rates listed in Table 8 after initial transient period. Fig. 12(b) shows the link utilization and buffer occupancy of the congested link (Link 34). Again, the GMM-bottleneck link is 100% utilized with small buffer requirements.

In summary, based on the simulation results in this section, we have demonstrated that our distributed algorithm achieves the GMM rate allocation policy in a regional enterprise network environment.

For a wide are network, the effectiveness of our simple distributed algorithm depends on careful system parameter tuning to minimize oscillations. Here, a more sophisticated distributed algorithm using per-flow state information [7] might provide better performance. But in a regional enterprise network environment, where implementation cost is critical, our simple algorithm is a viable solution ($O(1)$ storage requirements and computational complexity).

V. CONCLUSIONS

The main contributions of this work are the generalization of the theory of the classical max-min to include the minimum rate and peak rate constraints for each flow, and the development of a simple distributed algorithm to achieve the generalized max-min rate allocation policy. Simulation results based on several benchmark network configurations demonstrated the effective-

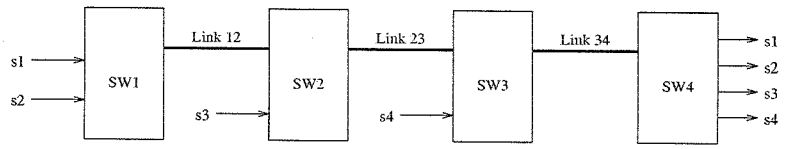

Fig. 11. A parking-lot network.

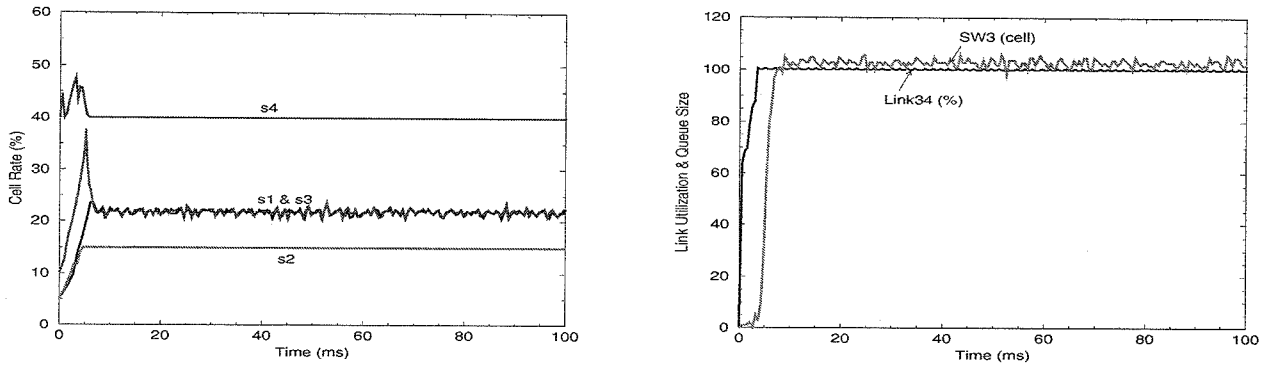

Fig. 12. (a) The cell rates of all flows; (b) the link utilization and queue size of the congested switch for the parking-lot network.

ness of our distributed algorithm.

APPENDIX

A proof of Theorem 2 is given as follows.

Proof: The existence part is proved by the construction of a rate vector through a centralized rate allocation algorithm (e.g., Algorithm 1) and show that the vector satisfies the max-min policy. Therefore, we only need to prove the uniqueness part here. Suppose that the rate vector r is max-min and we will show that there cannot exist another rate vector which is also max-min.

Assume that there is some other feasible rate vector $\hat{r} \neq r$ which is also max-min. Then there must exist a flow $s \in \mathcal{S}$ such that $\hat{r}^s > r^s$ (If $\hat{r}^s \leq r^s$ for every $s \in \mathcal{S}$, then some flow in $\hat{r}$ does not have a bottleneck link and the rate vector $\hat{r}$ cannot be max-min). For the rate vector r, at the bottleneck link ℓ for flow s, we have $F_\ell = C_\ell$ and $r^s \geq r^t$ for all t traversing ℓ. Since $\hat{r}^s > r^s$, for feasibility we must have some flow t with $\hat{r}^t < r^t$ at the bottleneck link ℓ for flow s. Thus the quantity

$$\delta = \min\{(\hat{r}^s - r^s),\ \ (r^t - \hat{r}^t)\}$$

is positive. Therefore, we can increase $\hat{r}^t$ by δ while decreasing the same amount of rate from flow s with $\hat{r}^s$ on the link ℓ traversed by s and t with $F_\ell = C_\ell$. We maintain feasibility without decreasing the rate of any flow p with $\hat{r}^p \leq \hat{r}^t$ and this contradicts the max-min definition of the rate vector $\hat{r}$. □

The following is a proof for Theorem 3.

Proof: To show the *"only if"* part, we suppose that the ABR-feasible rate vector r is GMM and assume that on the contrary that there exists some flow $s \in \mathcal{S}$ which has neither an GMM-bottleneck link with respect to r nor a rate assignment equal to its PCR. Then for every non-saturated link ℓ ($F_\ell < C_\ell$) traversed by s, we can increase r^s by an increment until it reaches the PCR of s or some link saturates, whichever is smaller. For every saturated link ℓ ($F_\ell = C_\ell$) traversed by s, if $T = \{t \mid r^t > \text{MCR}^t, t \text{ traversing } \ell\}$ is nonempty, there must exist a flow $p \in T, p \neq s$, such that $r^p > r^s$. Thus the quantity

$$\delta_\ell = \begin{cases} \min\{(C_\ell - F_\ell),\ \ (\text{PCR}^s - r^s)\} & \text{if } F_\ell < C_\ell, \\ \min\{(r^p - r^s),\ \ (r^p - \text{MCR}^p),\ \ (\text{PCR}^s - r^s)\} & \text{if } F_\ell = C_\ell \end{cases}$$

is positive. 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 flow r^p on the links ℓ traversed by s with $F_\ell = C_\ell$. We maintain ABR-feasibility without decreasing the rate of any flow t with $r^t \leq r^s$. This contradicts the GMM definition of the rate vector r.

For the proof of the *"if"* part of Theorem 3, we assume that each flow has either an GMM-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 flow s with $r^s < \text{PCR}^s$ while maintaining ABR-feasibility, we must decrease the rate of some flow p with $r^p > \text{MCR}^p$ and p traverses the GMM-bottleneck link ℓ of s (flow s must go through an GMM-bottleneck link since $r^s < \text{PCR}^s$ and we have $F_\ell = C_\ell$ by the definition of an GMM-bottleneck link). Since $r^s \geq r^p$ for all p in $T = \{t \mid r^t > \text{MCR}^t, t \text{ traversing } \ell\}$ by the definition of GMM-bottleneck link, the rate assignment for any flow $s \in \mathcal{S}$ with $r^s < \text{PCR}^s$ satisfies the definition for GMM rate allocation.
- *Case 2:* For any flow 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 flow s with $r^s = \text{PCR}^s$ satisfies the requirement for GMM.

Combining Cases 1 and 2, we have proved the *"if"* part of the theorem. □

The correctness proof of Algorithm 3 is given as follows.

Proof: Since 1) At least one of the following three events happens during an iteration of Algorithm 3: i) The rate of flows in u_1 reaches the rate of flows in u_2, which is MCR^t, $t \in u_2$, ii) Some link saturates, and iii) Some flow in u_1 reaches its PCR; and 2) the number of flows in the network $\mathcal{N}$ is a constant equal to $|\mathcal{S}|$, the algorithm terminates at most by $(2|\mathcal{S}| - 1)$ iterations (at most $(|\mathcal{S}| - 1)$ iterations for $(|\mathcal{S}| - 1)$ flows to reach the maximum MCR value among all flows and another $|\mathcal{S}|$ iterations for each flow to reach its GMM-bottleneck link rate or its PCR).

The correctness of this algorithm is proved by showing that each flow will either have some GMM-bottleneck link with respect to the final rate vector r or a rate assignment equal to its PCR when the algorithm terminates. Initially, the rate allocation of each flow $s \in S$ is $r^{s,(0)} = \text{MCR}^s$. During each iteration, an equal increment of rate is added to all flows in u_1 (each flow in u_1 has the same rate) and as a result one of the following three events listed above happens at the end of an iteration.

- *Case 1:* Suppose that the rate of flows in u_1 reaches the rate of flows in u_2 at the kth iteration. Then no flow is removed from $\mathcal{S}^{(k)}$ during this iteration. The flows in both u_1 and u_2 become the new u_1 at the $(k+1)$th iteration. The new u_2 at the $(k+1)$th iteration is u_3 from the kth iteration, etc. That is, the number of sorted sets m for the $(k+1)$th iteration is 1 less than that in the kth iteration. Case i can happen at most $(|\mathcal{S}| - 1)$ times in executing Algorithm 3 since the number of flows is a constant with $|\mathcal{S}|$.
- *Case 2:* Suppose that a link ℓ is saturated at the kth iteration, and $s \in \mathcal{S}^{(k)}$ traverses ℓ. We have $r^s \geq r^t$ for every t traversing ℓ such that $r^t > \text{MCR}^t$. That is, link ℓ is an GMM-bottleneck link with respect to r for flow s.
- *Case 3:* Suppose that flow s reaches its PCR at the kth iteration. Then $r^{s,(k)} = \text{PCR}^s$ and $r^{s,(k)}$ will not be increased further during future iterations.

As a result, upon termination of the algorithm, each flow either has some GMM-bottleneck link or a rate assignment equal to its PCR. By Theorem 3, the final rate vector r satisfies the GMM policy. □

REFERENCES

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

[2] ATM Forum Technical Committee, "Traffic management specification, version 4.0," *ATM Forum Contribution 96-0056.00*, Apr. 1996.

[3] D. Hughes, "Fair share in the context of MCR," *ATM Forum Contribution 94-0977*, Oct. 1994.

[4] N. Yin, "Max-min fairness vs. MCR guarantee on bandwidth allocation for ABR," in *Proc. IEEE ATM'96 Workshop*, San Francisco, CA, Aug. 1996.

[5] K.-Y. Siu and H.-Y. Tzeng, "Limits of performance in rate-based control schemes," *ATM Forum Contribution 94-1077*, Nov. 1994.

[6] H.-Y. Tzeng and K.-Y. Siu, "Comparison of performance among existing rate control schemes," *ATM Forum Contribution 94-1078*, Nov. 1994.

[7] A. Charny, D. Clark, and R. Jain, "Congestion control with explicit rate indication," in *Proc. IEEE ICC'95*, June 1995, pp. 1954–1963.

[8] K.-Y. Siu and H.-Y. Tzeng, "Intelligent congestion control for ABR service in ATM networks," *ACM SIGCOMM Computer Commun. Review*, vol. 24, no. 5, pp. 81–106, 1994.

[9] N. Yin and M. G. Hluchyj, "On closed-loop rate control for ATM cell relay networks," in *Proc. IEEE INFOCOM'94*, June 1994, pp. 99–108.

Yiwei Thomas Hou obtained his B.E. degree (*Summa Cum Laude*) from the City College of New York in 1991, the M.S. degree from Columbia University in 1993, and the Ph.D. degree from Polytechnic University, Brooklyn, New York, in 1997, all in Electrical Engineering. He was awarded a five-year National Science Foundation Graduate Research Traineeship for pursuing Ph.D. degree in high speed networking, and was recipient of the Alexander Hessel award for outstanding Ph.D. dissertation (1997–1998 academic year) from Polytechnic University. While a graduate student, he worked at AT&T Bell Labs, Murray Hill, New Jersey, during the summers of 1994 and 1995, on internetworking of IP/ATM networks; he conducted research at Lucent Technologies Bell Labs, Holmdel, New Jersey, during the summer of 1996, on fundamental problems on network traffic management.

Since September 1997, Dr. Hou has been a Research Scientist at Fujitsu Laboratories of America, Sunnyvale, California. He received Intellectual Property Contribution Award from Fujitsu Laboratories of America in 1999. His current research interests are in the areas of scalable architecture, protocols, and implementations for differentiated services Internet, terabit switching, and quality of service (QoS) support for multimedia over IP networks. Dr. Hou is a member of the IEEE, ACM, Sigma Xi, and New York Academy of Sciences.

Shivendra S. Panwar received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Kanpur, in 1981, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Massachusetts at Amherst, in 1983 and 1986, respectively. He joined the Department of Electrical Engineering at the Polytechnic University, Brooklyn, NY, and is now an Associate Professor. Since 1996, he has served as Director of the New York State Center for Advanced Technology in Telecommunications (CATT). His research interests include performance analysis and design of high speed networks. Dr. Panwar is a Senior Member of the IEEE and a member of Tau Beta Pi and Sigma Xi. He was co-editor of two books, *Network Management and Control, Vol. II*, and *Multimedia Communications and Video Coding*, both published by Plenum.

Henry Tzeng received his B.S. degree from the Tatung Institute of Technologies, Taiwan, Republic of China, in 1988, and his M.S. and Ph.D. degrees in Electrical Engineering from the University of California, Irvine, in 1993 and 1995, respectively. He was a recipient of the University of California Regent's Dissertation Fellowship in 1995 and the 1997 IEEE Browder J. Thompson Memorial Prize Award. Dr. Tzeng is currently Principle Engineer at Amber Networks Inc., Santa Clara, CA, and is working on high performance routing technologies for the Internet. Prior to joining Amber Networks, he was Member of Technical Staff at Lucent Technologies Bell Labs, Holmdel, NJ, during 1995 to 1999. Dr. Tzeng was a Guest Editor for the *IEEE Journal on Selected Areas in Communications* special issue on next generation IP switches and routers (June 1999).