A Generalized Max-Min Rate Allocation Policy and Its Distributed Implementation Using the ABR Flow Control Mechanism *

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

Abstract

We generalize the classical max-min rate allocation policy with the support of the minimum rate requirement and peak rate constraint for each connection. Since a centralized algorithm for the generalized max-min (GMM) rate allocation requires global information, which is difficult to maintain and manage in a large network, we develop a distributed protocol to achieve the GMM policy using the available bit rate (ABR) flow control mechanism. We give a proof that our distributed protocol converges to the GMM rate allocation through distributed and asynchronous iterations under any network configuration and any set of link distances.

1 Introduction

The classical max-min policy has been suggested by the ATM Forum as a network bandwidth sharing policy for ABR service [1, 2]. This works well when each ABR connection's minimum cell rate (MCR) is zero and peak cell rate (PCR) is greater than or equal to the link rate. But in a general setting of MCR/PCR for a connection, the classical max-min policy no longer suffices to determine a rate allocation since it does not support either MCR or PCR.

In this paper, we generalize the classical max-min policy with the support of a minimum rate requirement and a peak rate constraint for each connection.

Since a centralized algorithm for the generalized max-min (GMM) policy requires global information, which is difficult to maintain and manage in a large network, we design a distributed protocol to achieve the GMM policy using the ABR flow control mechanism. Our distributed protocol is a generalization of Charny's *Consistent Marking* technique, which was originally designed to achieve the classical max-min policy [4]. We give a correctness proof that our distributed protocol converges to the GMM rate allocation through distributed and asynchronous iterations. Our convergence proof gives a theoretical guarantee that the rate allocation by our distributed protocol converges to our GMM policy under *any* network configuration and *any* set of link distances.

The remainder of this paper is organized as follows. Section 2 presents the generalized max-min (GMM) policy with the support of a minimum rate requirement and a peak rate constraint for each connection. In Section 3, we present a distributed protocol to achieve the GMM policy; and in Section 4 we give a proof of its convergence. Section 5 shows simulation results of our distributed protocol. Section 6 concludes this paper and points out future research directions.

2 The Generalized Max-Min Rate Allocation Policy

In our model, a network $\mathcal{N}$ is characterized by interconnecting switches with a set of links $\mathcal{L}$. A session $s \in \mathcal{S}$ traverses one or more links in $\mathcal{L}$ and is allocated a specific rate r_s.[1] The aggregate allocated rate F_ℓ on link $\ell \in \mathcal{L}$ of the network is

$$F_\ell = \sum_{s \in \mathcal{S} \text{ traversing link } \ell} r_s .$$

Let C_ℓ be the capacity of link ℓ. A link ℓ is *saturated* or *fully utilized* if $F_\ell = C_\ell$. Let MCR_s and PCR_s be the minimum rate requirement and the peak rate constraint for each session $s \in \mathcal{S}$. We assume that the sum of all sessions' MCR traversing any link does not exceed the link's capacity. 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 $r = \{r_s \mid s \in \mathcal{S}\}$ is *ABR-feasible* if the following two constraints are satisfied: 1) $\text{MCR}_s \le r_s \le \text{PCR}_s$ for all $s \in \mathcal{S}$; and 2) $F_\ell \le C_\ell$ for all $\ell \in \mathcal{L}$.

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

Example 1 Peer-to-Peer Network
In this network configuration (Fig 1), the output port

*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.

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

0-7803-4383-2/98/$10.00 © 1998 IEEE.

link of SW1 (Link12) is the only potential bottleneck link. Assume that all links are of unit capacity. The MCR requirements and PCR constraints for all connections are listed in Table 1.

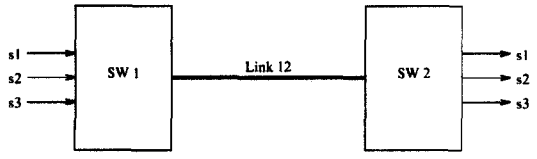

Figure 1: The peer-to-peer network configuration.

Session	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

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

The iterative steps to achieve the GMM rate allocation are listed below, with a graphical display shown in Fig. 2.

- Step 1: As shown in Fig. 2, we start the rate of each session with its MCR (shown in the darkest shaded areas in Fig. 2).
- Step 2: Since the rate of $s3$ (0.05) is the smallest among all sessions, 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.
- Step 4: Remove $s2$ (with a rate of 0.25) from 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 allocation is 0.40, 0.25, and 0.35 for $s1$, $s2$, and $s3$, respectively. □

The above example illustrates the fundamental concept of GMM policy, i.e. always *max*imize the *min*imum rate among all sessions (while satisfying each session's minimum rate requirement and peak rate constraint), which is the same concept as the classical max-min policy.

The iterative steps used in Example 1 for GMM rate allocation is characterized by the following algorithm, which can be applied to any network with an arbitrary number of connections.

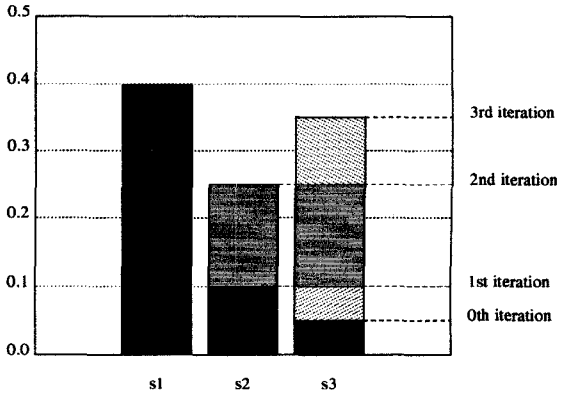

Figure 2: Graphical display of rate allocation for each session at each iteration under the GMM policy in the peer-to-peer network.

Algorithm 1 A Centralized Algorithm for the GMM Policy

1. Start the rate of each session with its MCR.
2. Sort all sessions in the order of increasing rate.
3. Increase the rate of the session with the smallest rate among all sessions until one of the following events takes place:
   - The rate of such a session reaches the second smallest rate among the sessions;
   - Some link saturates;
   - The session's rate reaches its PCR.
4. If some link saturates or the session's rate reaches its PCR in Step 3, remove the sessions that either traverse the saturated link or reach their PCRs, respectively, as well as the network capacity associated with such sessions from the network.
5. If there is no session left, the algorithm terminates; otherwise, go back to Step 3 for the remaining sessions and network capacity. □

Formally, the GMM rate allocation policy is defined as follows.

Definition 1 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 session $t \in \mathcal{S}$ such that $r_s \geq r_t$, and $r_t > \hat{r}_t$. □

We define a new notion of bottleneck link as follows.

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

Theorem 1 An ABR-feasible rate vector r is GMM if and only if each session has either a GMM-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 GMM rate allocation. □

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

In Example 1, Link 12 is a GMM-bottleneck link for both $s1$ and $s3$ (see Definition 2). On the other hand, $s1$ and $s3$ have different rate allocation (0.4 for $s1$ and 0.35 for $s3$). Thus, it is essential to have a precise definition of *GMM-bottleneck link rate* here.

Let $1^+\{\text{event A}\}$ be the indicator function with the following definition:

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

Definition 3 Given a GMM rate vector r, suppose that link $\ell \in \mathcal{L}$ is a GMM-bottleneck link with respect to r and let τ_ℓ denote the GMM-bottleneck link rate at link ℓ. Then τ_ℓ satisfies

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

where $\mathcal{U}_\ell$ denotes the set of sessions that are GMM-bottlenecked at link ℓ, and $\mathcal{Y}_\ell$ denotes the set of sessions that are either GMM-bottlenecked elsewhere or have a rate allocation equal to their PCRs and $r_\ell^i < \tau_\ell$ for $i \in \mathcal{Y}_\ell$. □

With the above definition, it is easy to show that in Example 1 the GMM-bottleneck link rate at Link 12 is 0.35.

Note 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 3 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 sessions bottlenecked at link ℓ. This is exactly the expression for the max-min bottleneck link rate at link ℓ.

It should be clear that by Definition 3 and Algorithm 1, the GMM rate allocation for a session $s \in \mathcal{S}$ can only be one of the following: 1) A rate equal to its MCR; or 2) A rate equal to its PCR; or 3) A rate equal to its GMM-bottleneck link rate.

The centralized algorithm for the GMM policy requires global information, which is difficult to maintain and manage in a large network. In the following sections, we design a distributed protocol to achieve the GMM policy and prove its convergence.

3 A Distributed Protocol

There have been extensive prior efforts on the design of distributed algorithms to achieve the classical max-min rate allocation. Early algorithms by Hayden [6], Jaffe [8], and Gafni [5] required synchronization of all nodes for each iteration. Mosely's work in [11] was the first asynchronous algorithm. Unfortunately, this algorithm could not offer satisfactory convergence performance. Later, Ramakrishnan *et al.* proposed to use a single bit to indicate congestion and achieve max-min [12]. Due to the binary nature of this algorithm, the source's rate exhibited oscillations. Recent interest in ABR service have led to many contributions to the design of distributed algorithm to achieve max-min [4, 9, 10, 13, 14, 15]. In particular, the algorithm by Charny *et al.* in [4] was one of the few algorithms that were proven to converge to max-min through distributed and asynchronous iterations. In this paper, we will make a generalization of Charny's *Consistent Marking* technique to design a distributed algorithm for our GMM rate allocation.

3.1 Distributed Control Mechanism

It should be clear that a distribute protocol achieving GMM rate allocation must have the cooperation between the sources and the network. In particular, such cooperation includes the following components: 1) Information exchange between a source and the network; and 2) Source rate adaptation upon receiving feedback from the network.

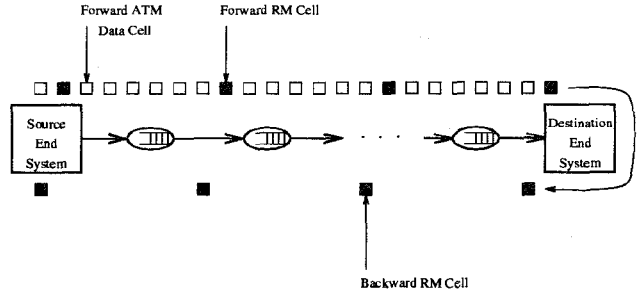

Figure 3: ABR flow control mechanism for a connection.

ABR flow control mechanism offers such a facility to achieve cooperation between a source and the network. As shown in Fig. 3, to achieve information exchange, Resource Management (RM) cells are inserted among data cells to exchange information between a source and the network. The source sets the fields in the forward RM cells to inform the network about the source's rate information (e.g. MCR, PCR, CCR). The network (switches)

set the fields in the backward RM cells (e.g. ER) to inform the source about available bandwidth. For source rate adaptation, the source adjusts its transmission rate upon receiving backward RM cells.

3.2 Charny's Work

Our distributed protocol for the GMM policy is motivated by the *Consistent Marking* technique by Charny *et al.* for the classical max-min policy [4]. In that algorithm, each switch monitors its traffic by keeping track of the state information of each traversing connection. Also, each output port of a switch maintains a variable called the *advertised rate* to calculate available bandwidth for each connection. When an RM cell arrives at the switch, the CCR value of the connection is stored in a VC table. If this CCR value is less than or equal to the current advertised rate, then the associated connection is assumed to be bottlenecked either at this link or elsewhere and a corresponding bit for this connection is marked at the VC table. Then the following equation is used to update the advertised rate:

$$\text{Advertised Rate} = \frac{C_\ell - \sum \text{Rates of marked connections}}{n_\ell - \sum \text{Marked connections}} \quad (1)$$

where C_ℓ and n_ℓ are the link capacity and the number of connections at link ℓ. Then the VC table is examined again. For each marked session, if its recorded CCR is larger than this newly calculated advertised rate, this session is then unmarked and the advertised rate is calculated again. The ER field of an RM cell is then set to the minimum of all advertised rates along its traversing links. Upon convergence, each session is allocated with a max-min rate and is marked along every link it traverses.

To extend Charny's algorithm for GMM, it is obvious the advertised rate calculation in (1) has to be modified to reflect the GMM-bottleneck link rate defined in Definition 3. However, with the newly defined GMM-bottleneck link rate, it is not clear how the marking should be done for each traversing session. For instance, in Example 1, if we mark a session when its CCR is less than or equal to the advertised rate as in Charny's technique, this will bring the advertised rate into a state of oscillation that will never converge!

A deeper look at Charny's original algorithm for max-min shows that *a session traversing its own max-min bottleneck link does not need to be marked at this link.* That is, at a saturated link, only sessions bottlenecked elsewhere need to be marked. In fact, this observation leads us to a fundamental generalization of Charny's Consistent Marking technique as well as its convergence proof, which we will elaborate in the following sections.

3.3 A Distributed Protocol for GMM

We first specify the end system behavior of our protocol, which conforms to the ABR framework in [1].

Algorithm 2 End System Behavior

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

Destination Behavior

- The destination returns every RM cell back towards the source upon receiving it. □

The switch maintains a table at each output port to keep track of the state information of each traversing VC (so-called per-VC accounting) and performs the switch algorithm (Algorithm 4) at this output port.

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

C_ℓ: Capacity of link ℓ, $\ell \in \mathcal{L}$.
RC_ℓ: Remaining Capacity variable at link ℓ used for μ_ℓ calculation in Algorithm 3.
$\mathcal{G}_\ell$: 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. $\mathcal{Y}_\ell = \{i \mid i \in \mathcal{G}_\ell \text{ and } b_\ell^i = 1\}$.
$\mathcal{U}_\ell$: Set of sessions unmarked at link ℓ, i.e. $\mathcal{U}_\ell = \{i \mid i \in \mathcal{G}_\ell \text{ and } b_\ell^i = 0\}$, and $\mathcal{Y}_\ell \cup \mathcal{U}_\ell = \mathcal{G}_\ell$.
μ_ℓ: Advertised rate at link ℓ, calculated as follows.

Algorithm 3 μ_ℓ Calculation

- if $n_\ell = 0$, then $\mu_\ell := C_\ell$;
- else if $n_\ell = |\mathcal{Y}_\ell|$, then $\mu_\ell := C_\ell - \sum_{i \in \mathcal{G}_\ell} r_\ell^i + \max_{i \in \mathcal{G}_\ell} r_\ell^i$;
- else {

  $RC_\ell := C_\ell - \sum_{i \in \mathcal{Y}_\ell} r_\ell^i$;
  if $(RC_\ell < \sum_{i \in \mathcal{U}_\ell} \text{MCR}^i)$ then $\mu_\ell := 0$;
  else /* i.e. $RC_\ell \geq \sum_{i \in \mathcal{U}_\ell} \text{MCR}^i$. */ {
  Sort unmarked sessions $s \in \mathcal{U}_\ell$ in increasing

order of their MCRs, i.e.
$\text{MCR}[1] \le \text{MCR}[2] \le \cdots \le \text{MCR}[|\mathcal{U}_\ell|]$;
$k := |\mathcal{U}_\ell|$; $\mu_\ell := \frac{RC_\ell}{k}$;
while ($\mu_\ell < \text{MCR}[k]$) {
$RC_\ell := RC_\ell - \text{MCR}[k]$;
$k := k - 1$;
$\mu_\ell := \frac{RC_\ell}{k}$;
}
}[2]
}

□

Each link $\ell \in \mathcal{L}$ is initialized with: $\mathcal{G}_\ell = \emptyset$; $n_\ell = 0$; $\mu_\ell = C_\ell$.

Algorithm 4 Switch Behavior

Upon the receipt of a forward RM(CCR, MCR, ER) cell from the source of session i {
if RM cell signals session exit[3]{
$\mathcal{G}_\ell := \mathcal{G}_\ell - \{i\}$; $n_\ell := n_\ell - 1$;
table_update();
}
if RM cell signals session initiation {
$\mathcal{G}_\ell := \mathcal{G}_\ell \cup \{i\}$; $n_\ell := n_\ell + 1$;
$b_\ell^i := 0$; $r_\ell^i := \text{CCR}$; $\text{MCR}^i := \text{MCR}$;
table_update();
}
else /* i.e. RM cell belongs to an onging session. */ {
$r_\ell^i := \text{CCR}$; if ($r_\ell^i < \mu_\ell$) then $b_\ell^i := 1$;
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};
Forward RM(CCR, MCR, ER) towards its source;
}

table_update()
{
rate_calculation_1: use Algorithm 3 to calculate advertised rate μ_ℓ^1;
Unmark any marked session $i \in \mathcal{G}_\ell$ at link ℓ with $r_\ell^i \ge \mu_\ell^1$;
rate_calculation_2: use Algorithm 3 to calculate advertised rate μ_ℓ;
if ($\mu_\ell < \mu_\ell^1$), then {
Unmark any marked session $i \in \mathcal{G}_\ell$ at link ℓ with $r_\ell^i \ge \mu_\ell$;
rate_calculation_3: use Algorithm 3 to calculate advertised rate μ_ℓ again;
}[4]
}

□

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

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

Remark 1 Unlike Charny's technique where a session is marked if its rate is less than or equal to the advertised rate, we mark a session only when its rate is strictly less than the advertised rate. A small modification as it may seem to be, this new marking criterion brings a whole new marking property for sessions upon convergence. In particular, a session traversing its own GMM-bottleneck link will *not* be marked at such a link upon convergence. In fact, this is the key to resolve the difficulty of marking sessions that are GMM-bottlenecked at the same link but with different rates (e.g. 0.4 for $s1$ and 0.35 for $s3$ in Example 1). In conjunction with the GMM-bottleneck link rate definition and advertised rate calculation, this new marking technique leads to a fundamental generalization of Charny's Consistent Marking technique, as we will show in detail in the following section on the proof of convergence. □

4 Proof of Convergence

The proof of convergence of our distributed protocol is based on a sequence of lemmas. We first give the following definition for *marking-consistent*, which is a generalization of Charny's definition in [4].

Definition 4 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 $r_\ell^i < \mu_\ell$ for every session $i \in \mathcal{Y}_\ell$. □

The following is a key lemma in our convergence proof.

[2]The combined steps in the bracket for "else" are equivalent to find the GMM-bottleneck link rate μ_ℓ for the set of unmarked sessions $\mathcal{U}_\ell$ such that $\mu_\ell \cdot \sum_{i \in \mathcal{U}_\ell} 1^+\{\text{MCR}^i \le \mu_\ell\} + \sum_{i \in \mathcal{U}_\ell} \text{MCR}^i \cdot 1^+\{\text{MCR}^i > \mu_\ell\} = RC_\ell$. In the special case when $\text{MCR}^i = 0$ for every $i \in \mathcal{U}_\ell$, $\mu_\ell = \frac{RC_\ell}{|\mathcal{U}_\ell|}$, i.e. the max-min share rate.

[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 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. See the proof of Lemma 1 for a unique case where μ_ℓ by rate_calculation_2 may be less than μ_ℓ^1 and another round of unmarking and rate_calculation_3 is necessary.

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. □

Proof of Lemma 1: Let $\mathcal{Y}_\ell$ and $\mathcal{U}_\ell$ be the set of marked and unmarked sessions at link ℓ just before rate_calculation_1 is performed, respectively; μ_ℓ^1 be the result for the advertised rate by rate_calculation_1 in function table_update(); $\mathcal{Z}_\ell \subseteq \mathcal{Y}_\ell$ be the set of sessions with $r_\ell^i \geq \mu_\ell^1$, $i \in \mathcal{Z}_\ell$ and therefore, are unmarked by the unmarking operation after rate_calculation_1 in function table_update(); μ_ℓ be the result for advertised rate by rate_calculation_2 in function table_update().

- *Case 1:* If not all sessions in $\mathcal{G}_\ell$ are marked before rate_calculation_1, i.e. $\mathcal{Y}_\ell \neq \mathcal{G}_\ell$, then we have the following two scenarios.
  Subcase A: During rate_calculation_1, if $C_\ell - \sum_{i\in\mathcal{Y}_\ell} r_\ell^i < \sum_{i\in\mathcal{U}_\ell} \mathrm{MCR}^i$, then $\mu_\ell^1 = 0$. Thus, every session $i \in \mathcal{Y}_\ell$ will be unmarked by the unmarking operation and μ_ℓ calculated by rate_calculation_2 satisfies

$$\mu_\ell \cdot \sum_{i\in\mathcal{G}_\ell} 1^+\{\mathrm{MCR}^i \leq \mu_\ell\} + \sum_{i\in\mathcal{G}_\ell} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \mu_\ell\} = C_\ell$$

  and $C_\ell \geq \sum_{i\in\mathcal{G}_\ell} \mathrm{MCR}^i$ (by call admission control). Therefore, $\mu_\ell \geq \mu_\ell^1 = 0$ and marking-consistent property trivially holds.
  Subcase B: During rate_calculation_1 for μ_ℓ^1, if

$$C_\ell - \sum_{i\in\mathcal{Y}_\ell} r_\ell^i \geq \sum_{i\in\mathcal{U}_\ell} \mathrm{MCR}^i \tag{2}$$

  then μ_ℓ^1 satisfies

$$\begin{aligned}\mu_\ell^1 \cdot \sum_{i\in\mathcal{U}_\ell} 1^+\{\mathrm{MCR}^i \leq \mu_\ell^1\} &+ \sum_{i\in\mathcal{U}_\ell} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \mu_\ell^1\} \\ &= C_\ell - \sum_{i\in\mathcal{Y}_\ell} r_\ell^i. \end{aligned} \tag{3}$$

  After unmarking $\mathcal{Z}_\ell \subseteq \mathcal{Y}_\ell$ with $r_\ell^i \geq \mu_\ell^1$, $i \in \mathcal{Z}_\ell$, in function table_update(), we have

$$\begin{aligned} C_\ell - \sum_{i\in(\mathcal{Y}_\ell - \mathcal{Z}_\ell)} r_\ell^i &= C_\ell - \sum_{i\in\mathcal{Y}_\ell} r_\ell^i + \sum_{i\in\mathcal{Z}_\ell} r_\ell^i \\ &\geq \sum_{i\in\mathcal{U}_\ell} \mathrm{MCR}^i + \sum_{i\in\mathcal{Z}_\ell} \mathrm{MCR}^i \\ &= \sum_{i\in(\mathcal{U}_\ell\cup\mathcal{Z}_\ell)} \mathrm{MCR}^i \end{aligned}$$

  The inequality holds by (2) and by Fact 1, $\sum_{i\in\mathcal{Z}_\ell} r_\ell^i \geq \sum_{i\in\mathcal{Z}_\ell} \mathrm{MCR}^i$. In rate_calculation_2 for μ_ℓ, we have

$$\begin{aligned}\mu_\ell \cdot \sum_{i\in(\mathcal{U}_\ell\cup\mathcal{Z}_\ell)} 1^+\{\mathrm{MCR}^i \leq \mu_\ell\} &+ \sum_{i\in(\mathcal{U}_\ell\cup\mathcal{Z}_\ell)} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \mu_\ell\} \\ &= C_\ell - \sum_{i\in(\mathcal{Y}_\ell - \mathcal{Z}_\ell)} r_\ell^i. \end{aligned} \tag{4}$$

  But by (3),

$$\begin{aligned} C_\ell - \sum_{i\in(\mathcal{Y}_\ell - \mathcal{Z}_\ell)} r_\ell^i &= (C_\ell - \sum_{i\in\mathcal{Y}_\ell} r_\ell^i) + \sum_{i\in\mathcal{Z}_\ell} r_\ell^i \\ &= \mu_\ell^1 \cdot \sum_{i\in\mathcal{U}_\ell} 1^+\{\mathrm{MCR}^i \leq \mu_\ell^1\} \\ &\quad + \sum_{i\in\mathcal{U}_\ell} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \mu_\ell^1\} \\ &\quad + \sum_{i\in\mathcal{Z}_\ell} r_\ell^i. \end{aligned} \tag{5}$$

  Since $r_\ell^i \geq \mu_\ell^1$ and $r_\ell^i \geq \mathrm{MCR}^i$ for $i \in \mathcal{Z}_\ell$, to have (4) equal to (5), we must have $\mu_\ell \geq \mu_\ell^1$. That is, μ_ℓ calculated by rate_calculation_2 is greater than or equal to μ_ℓ^1 by rate_calculation_1. Since $r_\ell^i < \mu_\ell^1$ for $i \in (\mathcal{Y}_\ell - \mathcal{Z}_\ell)$, and $\mu_\ell^1 \leq \mu_\ell$, the marking of these sessions continues to be marking-consistent after rate_calculation_2 is performed.

- *Case 2:* If all sessions in $\mathcal{G}_\ell$ are marked before rate_calculation_1, i.e. $\mathcal{Y}_\ell = \mathcal{G}_\ell$, we have two scenarios. Let the RM cell for which the switch algorithm is performed belong to session $s \in \mathcal{S}$.
  Subcase A: If session s was not marked before the RM cell's arrival at link ℓ and is marked because of this RM cell's arrival with $r_\ell^s = \mathrm{CCR} < \mu_\ell$, where μ_ℓ was calculated by the switch algorithm for the previous traversing RM cell and satisfies

$$\mu_\ell = C_\ell - \sum_{i\in\mathcal{G}_\ell, i\neq s} r_\ell^i.$$

  After marking $b_\ell^s = 1$, we have

$$C_\ell - \sum_{i\in\mathcal{G}_\ell} r_\ell^i > 0. \tag{6}$$

  During rate_calculation_1 for μ_ℓ^1:

$$\mu_\ell^1 = C_\ell - \sum_{i\in\mathcal{G}_\ell} r_\ell^i + \max_{i\in\mathcal{G}_\ell} r_\ell^i.$$

  With (6), we have

$$\mu_\ell^1 > \max_{i\in\mathcal{G}_\ell} r_\ell^i \geq r_\ell^p \quad \text{for every session } p \in \mathcal{G}_\ell.$$

So all sessions in $\mathcal{G}_\ell$ will remain marked after the unmarking operation. Therefore, μ_ℓ calculated by rate_calculation_2 will be the same as μ_ℓ^1 and the marking of all sessions is marking-consistent.
Subcase B: If session s was already marked before this RM cell arriving at link ℓ, the arrival of this RM cell will not change the advertised rate μ_ℓ if the CCR in this RM cell is the same as r_ℓ^s in the current VC table at the switch. On the other hand, if the new CCR is different from the recorded CCR for this session in the VC table, r_ℓ^s will be updated with this new CCR value. During rate_calculation_1 for μ_ℓ^1, we have

$$\mu_\ell^1 = C_\ell - \sum_{i \in \mathcal{G}_\ell} r_\ell^i + \max_{i \in \mathcal{G}_\ell} r_\ell^i.$$

Again, let $\mathcal{Z}_\ell \subseteq \mathcal{Y}_\ell$ denote the set of sessions with $r_\ell^i \geq \mu_\ell^1$, $i \in \mathcal{Z}_\ell$ and therefore, are unmarked by the unmarking operation after rate_calculation_1 in function table_update().

If $\mathcal{Z}_\ell = \emptyset$, i.e. no session is unmarked, then μ_ℓ calculated by rate_calculation_2 will be the same as μ_ℓ^1 and all sessions will remain marking-consistent.

If $\mathcal{Z}_\ell \neq \emptyset$, then the set of sessions in $\mathcal{Z}_\ell$ will be unmarked since

$$r_\ell^i \geq \mu_\ell^1, \quad i \in \mathcal{Z}_\ell. \tag{7}$$

This is the only situation where μ_ℓ calculated by rate_calculation_2 may be less than μ_ℓ^1. If $\mu_\ell < \mu_\ell^1$, then we will perform another around of unmarking and μ_ℓ calculation (rate_calculation_3). It should be clear that the combined steps of rate_calculation_2, unmarking, and rate_calculation_3 here are equivalent to Case 1. Thus, μ_ℓ calculated by rate_calculation_3 is greater than or equal to μ_ℓ calculated by rate_calculation_2 and the marking of sessions is marking-consistent. □

Lemma 2 Let α_ℓ be defined as

$$\alpha_\ell \cdot \sum_{i \in \mathcal{G}_\ell} 1^+\{\mathrm{MCR}^i \leq \alpha_\ell\} + \sum_{i \in \mathcal{G}_\ell} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \alpha_\ell\} = C_\ell$$

for every $\ell \in \mathcal{L}$. There exists some time t_0 such that for $t \geq t_0$,

$$\mu_\ell \geq \alpha_\ell$$

for every $\ell \in \mathcal{L}$. □

Proof of Lemma 2: Let time t_0 be the time immediately after the switch algorithm is performed for an RM cell at link ℓ and $\mathcal{Y}_\ell$ and $\mathcal{U}_\ell$ denote the set of marked and unmarked sessions at link ℓ, respectively. By Lemma 1, the marking of sessions at link ℓ is marking-consistent. That is, every marked session i at link ℓ satisfies $r_\ell^i < \mu_\ell$.

- *Case 1:* If some sessions in $\mathcal{G}_\ell$ are not marked, i.e. $\mathcal{Y}_\ell \neq \mathcal{G}_\ell$, then

$$\mu_\ell \cdot \sum_{i \in \mathcal{U}_\ell} 1^+\{\mathrm{MCR}^i \leq \mu_\ell\} + \sum_{i \in \mathcal{U}_\ell} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \mu_\ell\} = C_\ell - \sum_{i \in \mathcal{Y}_\ell} r_\ell^i$$

Therefore,

$$\begin{aligned} \sum_{i \in \mathcal{Y}_\ell} r_\ell^i + \mu_\ell \cdot \sum_{i \in \mathcal{U}_\ell} 1^+\{\mathrm{MCR}^i \leq \mu_\ell\} & \\ + \sum_{i \in \mathcal{U}_\ell} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \mu_\ell\} &= C_\ell \\ &= \alpha_\ell \cdot \sum_{i \in \mathcal{G}_\ell} 1^+\{\mathrm{MCR}^i \leq \alpha_\ell\} \\ &\quad + \sum_{i \in \mathcal{G}_\ell} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \alpha_\ell\} \end{aligned}$$

Since $r_\ell^i < \mu_\ell$ for $i \in \mathcal{Y}_\ell$, we must have $\mu_\ell \geq \alpha_\ell$ (the equality holds only when $\mathcal{Y}_\ell = \emptyset$).

- *Case 2:* If all sessions in $\mathcal{G}_\ell$ are marked, i.e. $\mathcal{Y}_\ell = \mathcal{G}_\ell$, there are two possible scenarios.
Subcase A: Suppose $\max_{i \in \mathcal{G}_\ell} r_\ell^i \geq \alpha_\ell$. Since $\mu_\ell > \max_{i \in \mathcal{G}_\ell} r_\ell^i$, we have $\mu_\ell \geq \alpha_\ell$.
Subcase B: If $\max_{i \in \mathcal{G}_\ell} r_\ell^i < \alpha_\ell$, then for every session $i \in \mathcal{G}_\ell$, $r_\ell^i < \alpha_\ell$. Let session $p \in \mathcal{S}$ be the session such that $r_\ell^p = \max_{i \in \mathcal{G}_\ell} r_\ell^i$. Then

$$\begin{aligned} \mu_\ell &= C_\ell - \sum_{i \in \mathcal{G}_\ell} r_\ell^i + \max_{i \in \mathcal{G}_\ell} r_\ell^i = C_\ell - \sum_{i \in \mathcal{G}_\ell,\, i \neq p} r_\ell^i \\ &= \Big(\alpha_\ell \cdot \sum_{i \in \mathcal{G}_\ell} 1^+\{\mathrm{MCR}^i \leq \alpha_\ell\} \\ &\quad + \sum_{i \in \mathcal{G}_\ell} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \alpha_\ell\}\Big) - \sum_{i \in \mathcal{G}_\ell,\, i \neq p} r_\ell^i \\ &\geq \alpha_\ell \end{aligned}$$

The last inequality holds because

$$\alpha_\ell \cdot \sum_{i \in \mathcal{G}_\ell} 1^+\{\mathrm{MCR}^i \leq \alpha_\ell\} + \sum_{i \in \mathcal{G}_\ell} \mathrm{MCR}^i \cdot 1^+\{\mathrm{MCR}^i > \alpha_\ell\} \geq \alpha_\ell |\mathcal{G}_\ell|$$

and $\sum_{i \in \mathcal{G}_\ell,\, i \neq p} r_\ell^i \leq \alpha_\ell(|\mathcal{G}_\ell| - 1)$. □

Let M be the total number of iterations needed to execute Algorithm 1. We have $M \leq (2|\mathcal{S}| - 1)$, where $|\mathcal{S}|$ is the total number of sessions in the network [7]. During each iteration of Algorithm 1, there are three types of events: i) The rate of the session with the smallest rate reaches the rate of the session with the second smallest rate; ii) Some link saturates; and iii) Some session

reaches its PCR. In the worst case, a type i event can take at most $(|\mathcal{S}| - 1)$ iterations, in which case each session has a different MCR and $(|\mathcal{S}|-1)$ iterations will bring the rates of all sessions to the same rate of $\max_{p\in\mathcal{S}} \text{MCR}^p$. Note that there is no session removal during a type i event and the rate allocation for each session is temporary. On the other hand, type ii and iii iterations give a permanent rate assignment to some session and such a session is removed from future iterations. In the following, we will focus only on type ii and iii iterations and index such iterations as $1, \ldots, N$, where N denotes the total number of type ii and type iii iterations in executing Algorithm 1. We have shown in [7] that $N \leq |\mathcal{S}|$.

Let $\mathcal{S}_i$ be the set of sessions being removed at the end of the ith iteration, where i is the newly indexed iteration when we consider only type ii and iii iterations of Algorithm 1, $1 \leq i \leq N$. Sessions in $\mathcal{S}_i$ have reached their GMM rates. Let $\mathcal{L}_i$, $1 \leq i \leq N$ be the set of links traversed by sessions in $\mathcal{S}_i$. Note that $\mathcal{S}_1$, $\mathcal{S}_2$, ..., $\mathcal{S}_N$ are mutually exclusive and the sum of $\mathcal{S}_1$, $\mathcal{S}_2$, ..., $\mathcal{S}_N$ is $\mathcal{S}$, while $\mathcal{L}_1$, $\mathcal{L}_2$, ..., $\mathcal{L}_N$ may be mutually inclusive. That is, there may be links belonging to both $\mathcal{L}_i$ and $\mathcal{L}_{i+1}$. This happens when some session in $\mathcal{S}_i$ reaches its PCR before saturating link $\ell \in \mathcal{L}_i$ at the ith iteration and link $\ell \in \mathcal{L}_i$ becomes part of $\mathcal{L}_{i+1}$ at the $(i+1)$th iteration.

Let τ_i, $1 \leq i \leq N$ be defined as follows:

$$\tau_i = \max_{s\in\mathcal{S}_i} r_\ell^s \text{ such that } r_\ell^s > \text{MCR}^s,\ 1 \leq i \leq N.$$

That is, r_ℓ^s is either the PCR or the GMM-bottleneck link rate of some session $s \in \mathcal{S}_i$. By the operation of Algorithm 1, we have $\tau_1 < \tau_2 < \ldots < \tau_N$.

The following lemma states the inequality between the advertised rate μ_ℓ and τ_1 on every link $\ell \in \mathcal{L}$ in the network.

Lemma 3 Let t_0 and α_ℓ be defined as in Lemma 2.
i) If $\tau_1 = \alpha_\ell \leq \text{PCR}^s$ for $s \in \mathcal{S}_1$, i.e., the GMM-bottleneck link rate is reached before some session $s \in \mathcal{S}_1$ reaches its PCR, then for any $t > t_0$,

$$\mu_\ell \geq \tau_1 \quad \text{for every } \ell \in \mathcal{L}_1;$$
$$\mu_\ell > \tau_1 \quad \text{for every } \ell \in (\mathcal{L} - \mathcal{L}_1).$$

ii) If $\tau_1 = \text{PCR}^s < \alpha_\ell$ for $s \in \mathcal{S}_1$, i.e., some session $s \in \mathcal{S}_1$ reaches its PCR before the GMM-bottleneck link rate is reached, then for any $t > t_0$,

$$\mu_\ell > \tau_1 \quad \text{for every } \ell \in \mathcal{L}.$$

□

For a proof of Lemma 3, see [7].

The following lemma gives a base case for induction.

Lemma 4 There exists a $T_1 \geq 0$ such that:
i) If $\tau_1 = \alpha_\ell \leq \text{PCR}^s$ for $s \in \mathcal{S}_1$, i.e., the GMM-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 $\text{ER} = \max\{\tau_1,\ \text{MCR}\}$.
3. The ACR at source for every session $i \in \mathcal{S}_1$ satisfies $\text{ACR} = \max\{\tau_1,\ \text{MCR}\}$.
4. $r_\ell^i = \max\{\tau_1,\ \text{MCR}\}$ for every session $i \in \mathcal{S}_1$ and every link ℓ traversed by session $i \in \mathcal{S}_1$;
   $b_\ell^i = 1$ for every session with $r_\ell^i = \tau_1$, $i \in \mathcal{S}_1$ and every traversing link ℓ, *except* at its GMM-bottleneck link $\ell \in \mathcal{L}_1$.
5. The ER field of every returning RM cell of session $j \in (\mathcal{S} - \mathcal{S}_1)$ satisfies $\text{ER} > \tau_1$.
6. The ACR at source for every session $j \in (\mathcal{S} - \mathcal{S}_1)$ satisfies $\text{ACR} > \tau_1$.
7. The recorded CCR of session $j \in (\mathcal{S} - \mathcal{S}_1)$ satisfies $r_\ell^j > \tau_1$ at every link ℓ traversed by session j.

ii) If $\tau_1 = \text{PCR}^s < \alpha_\ell$ for $s \in \mathcal{S}_1$, i.e., some session $s \in \mathcal{S}_1$ reaches its PCR before the GMM-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.

□

For a proof of Lemma 4, see [7]. Note that Lemma 4 states that not only session $p \in \mathcal{S}_1$ has reached its GMM rate of $\max\{\tau_1,\ \text{MCR}^p\}$ (in case i) or PCR^p (in case ii), but that its rate will *never* change and such a session is marked with the following property:

- If $r_\ell^p = \text{MCR}^p$ (case i), then session p is not marked at its GMM-bottleneck link but may be marked at other links it traverses;
- If $r_\ell^p = \tau_1$ (case i), then session p is marked at all of its traversing links except its GMM-bottleneck link;

- If $r_\ell^p = \text{PCR}^p$ (case ii), then session p is marked at every link it traverses.

The result of Lemma 4 is used as the base case for induction on the index i of $\mathcal{S}_i$. It can be shown that if for some $1 \leq i \leq N-1$, there exists a $T_i \geq 0$ such that statements as the ones in Lemma 4 hold, then there exists a $T_{i+1} \geq 0$ such that for $t \geq T_{i+1}$, all statements also hold for $i+1$ [7]. This induction leads to the following main result.

Theorem 3 After the number of active sessions in the network stabilizes, the rate allocation for each session by our distributed protocol converges to the GMM rate allocation. □

It has also been shown in [7] that an upper bound for the convergence time to the final GMM rate allocation by our distributed protocol from the time when the number of active sessions in the network stabilizes is given by $2.5|\mathcal{S}|D$, where $|\mathcal{S}|$ is the total number of sessions in the network and D is an upper bound for the round-trip delay among all sessions.

5 Simulation Results

Our work in Section 4 gives a correctness proof that our distributed protocol in Section 3.3 converges to the GMM rate allocation through distributed and asynchronous iterations. This gives us a theoretical guarantee that our distributed protocol converges to the GMM policy under *any* network configuration and *any* set of link distances. In this section, we perform simulations to demonstrate the convergence property of our distributed protocol. Due to the paper length constraint, we will only show simulations on the generic fairness network configuration (Fig. 4). We refer interested readers to [7] for simulations on many other network configurations.

As shown in Fig. 4, the specific generic fairness network configuration that we use consists of 5 ATM switches connected in a chain with 6 session paths traversing these switches and sharing link capacity [3].

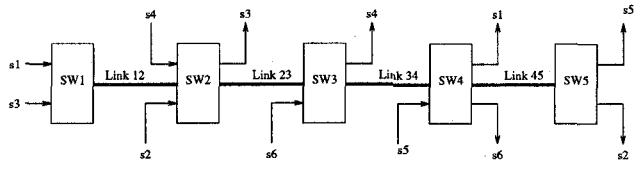

Figure 4: The generic fairness network configuration.

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

Session	MCR	PCR	GMM Rate Allocation
$s1$	0.05	0.50	0.35
$s2$	0.10	0.25	0.25
$s3$	0.15	1.00	0.65
$s4$	0.05	0.15	0.15
$s5$	0.35	1.00	0.75
$s6$	0.40	0.60	0.40

Table 2: MCR requirement, PCR constraint, and GMM rate allocation of each session for the generic fairness network configuration.

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

Table 3: Simulation parameters.

Table 2 lists the MCR requirement, PCR constraint and GMM rate allocation for each session.

Table 3 lists the parameters used in our simulation. The link speed 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.[5] 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.

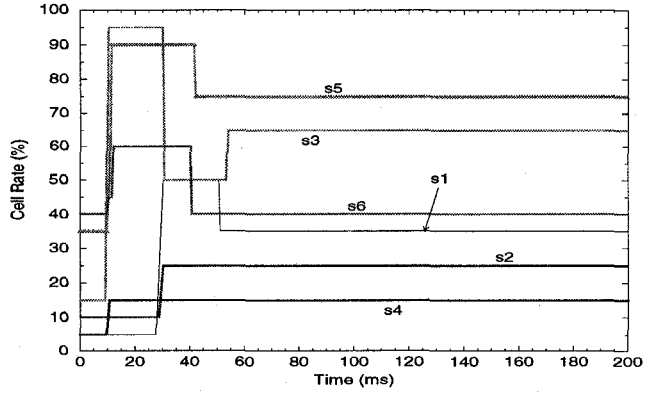

Figure 5: The cell rates of all connections for the generic fairness network configuration.

Fig. 5 shows the ACR at source for each session under our distributed protocol. The cell rates shown in the plot

[5] This will ensure that the potential buffer build up during transient period will be eventually emptied upon convergence.

are normalized with respect to the targeted link capacity C_ℓ (142.5 Mbps) for easy comparison with those values obtained with our centralized algorithm for GMM policy under unit link capacity (Table 2). Each session starts to transmit at its MCR. After initial iterations, we see that the cell rate of each session converges to the GMM rate allocation listed in Table 2. Here, the maximum round trip time (RTT) among all sessions is 30 ms ($s1$ and $s2$) and its takes less than 2 RTT (60 ms) for our algorithm to converge.

6 Concluding Remarks

The contributions of this work are three-fold. First, we generalized the classical max-min policy to include a minimum rate requirement and a peak rate constraint for each connection by extending the key concept of max-min, i.e., *max*imize the *min*imum rate among all connections. Secondly, we designed a distributed protocol with the aim of achieving the GMM rate allocation by making a generalization of Charny's Consistent Marking technique. Thirdly, and most importantly, we gave a proof that our distributed protocol converges to the GMM rate allocation through distributed and asynchronous iterations. Our proof provides a theoretical guarantee that our distributed protocol converges to the GMM rate allocation under any network configuration and any set of link distances.

Our future work will focus on other issues in our distributed protocol. One challenging problem is to reduce the storage and computational complexity of our switch algorithm and yet be able to have a rigorous proof of the algorithm's convergence. Other issues include system transient behavior, buffer requirements, and the rate of convergence, which all need to be carefully investigated before we deploy a distributed protocol.

References

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

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

[3] 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, Mar./Apr. 1995, pp.25-39.

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

[5] E. M. Gafni, "The Integration of Routing and Flow Control for Voice and Data in a Computer Communication Network," *Ph.D. Thesis,* Dept. of Elec. Eng. and Comp. Sci., MIT, Cambridge, MA, August 1982.

[6] H. P. Hayden, "Voice Flow Control in Integrated Packet Networks," *M.S. Thesis,* Dept. of Elec. Eng. and Comp. Sci., MIT, Cambridge, MA, June 1981.

[7] Y. T. Hou, H. Tzeng, and S. S. Panwar, "Generalized Max-Min Fairness for ABR Service: Theory and Implementation," *Tech. Report CATT 97-109,* Center for Advanced Technology in Telecommunications, Polytechnic University, Brooklyn, NY, Feb. 24, 1997.

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

[9] R. Jain, *et al.*, "ERICA Switch Algorithm: A Complete Description," *ATM Forum Contribution, 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),* Sept. 1995, pp.143-154.

[11] J. Mosely, "Asynchronous Distributed Flow Control Algorithms," *Ph.D. Thesis,* Dept. of Elec. Eng. and Comp. Sci., MIT, Cambridge, MA, June 1984.

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

[13] L. Roberts, "Enhanced PRCA (Proportional Rate Control Algorithm," *ATM Forum Contribution 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.