A Generalized Max-Min Network Capacity Assignment Policy with a Simple ABR Implementation for an ATM LAN *

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

Abstract

We introduce the generalized max-min (GMM) rate allocation policy, which is a direct generalization of the classical max-min policy with the support of both the minimum rate requirement and the peak rate constraint for each connection. A centralized algorithm is presented to compute network-wide bandwidth allocation to achieve this policy. Furthermore, a simple distributed algorithm with the aim of achieving the GMM policy is developed in the context of the ATM Forum ABR traffic management framework. The effectiveness of this distributed ABR algorithm is demonstrated by simulation results based on the benchmark network configurations suggested by the ATM Forum.

1 Introduction

The classical max-min policy has been widely used as an optimal rate allocation policy [2]. For example, the ATM Forum traffic management group has suggested the use of the max-min policy for available bit rate (ABR) service [1]. There have been extensive prior efforts to design distributed algorithms to achieve the max-min policy [3, 8, 9].

The classical max-min policy does not support the minimum rate requirement and the peak rate constraint for each connection. To address this issue, an MCR-offsetted and an MCR-weighted version of max-min policies were proposed in [6, 12].

In this paper, we introduce the generalized max-min (GMM) policy, which makes a direct generalization of the classical max-min policy by using its key concept, i.e. *max*imize the rate of the session with the *min*imum rate. Our policy supports both the minimum rate requirement and the peak rate constraint for each connection. We also present a centralized algorithm to achieve the GMM policy in any network topology with an arbitrary number of connections.

Then we move on to the design of distributed protocol to achieve the GMM policy for ATM ABR service. Our ABR algorithm is based on the *Intelligent Marking* technique by Siu and Tzeng [10, 11], which achieves the classical max-min rates without minimum rate and peak rate support. We make a simple extension of this technique to achieve the GMM policy.

The remainder of this paper is organized as follows. In Section 2, we define the generalized max-min (GMM) policy. We also present a centralized algorithm to achieve the GMM policy. In Section 3, we develop a distributed algorithm to achieve the GMM policy in the context of the ATM Forum ABR traffic management framework. In Section 4, we present simulation results to show the effectiveness of our ABR algorithm on a few benchmark network configurations suggested by the ATM Forum. Section 5 concludes this paper.

2 The Generalized Max-Min Policy

In this section, we present the theory of the generalized max-min (GMM) rate allocation policy. In our model, a network $\mathcal{N}$ is characterized by a set of links $\mathcal{L}$ and sessions $\mathcal{S}$.[1] Each session $s \in \mathcal{S}$ traverses one or more links in $\mathcal{L}$ and is allocated a specific rate r_s. The (aggregate) allocated rate F_ℓ on link $\ell \in \mathcal{L}$ of the network is

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

Let C_ℓ be the capacity (maximum allowable bandwidth) of link ℓ. A link ℓ is *saturated* or *fully utilized* if $F_\ell = C_\ell$. Let MCR_s and PCR_s be the minimum rate requirement and the peak rate constraint for each session $s \in \mathcal{S}$. For the sake of feasibility, we assume that the sum of all the sessions' MCR requirements traversing any link does not exceed the link's capacity, i.e.

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

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

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

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

Before we give a definition for the GMM policy, we give the following centralized algorithm for the GMM policy.

*This work is supported in part by the New York State Center for Advanced Technology in Telecommunications (CATT), Polytechnic University, Brooklyn, NY.

†Y. T. Hou is a Ph.D. candidate under the National Science Foundation Graduate Research Traineeship Program at Polytechnic University, Brooklyn, NY.

‡H. Tzeng is with Bell Labs, Lucent Technologies, Holmdel, NJ.

§S. S. Panwar is on the faculty of the Dept. of Electrical Engineering, Polytechnic University, Brooklyn, NY.

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

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 this session reaches the second smallest rate among all sessions;
   - Some link saturates;
   - The rate of this session reaches its PCR.
4. If some link saturates or such session reaches its PCR in Step 3, remove the sessions that either traverse this 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. □

We use the following simple example to illustrated how to use the above centralized algorithm to allocate network bandwidth for the GMM policy.

Example 1 Peer-to-Peer Configuration

In this network configuration (Fig. 1), the output port link of SW1 (Link12) is the only potential bottleneck link for all sessions. Assume that all links are of unit capacity. The MCR requirement and PCR constraint for each session are listed in Table 1.

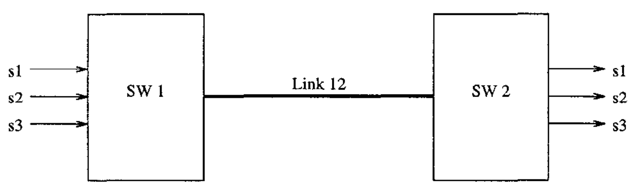

Figure 1: The peer-to-peer network configuration.

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 following steps describe how to use the centralized algorithm to allocate bandwidth for each session at each iteration, with a graphical display of the iterations in Fig. 2.

- Step 1: We start the rate allocation for each session with its MCR requirement (shown in the darkest shaded areas in Fig. 2).

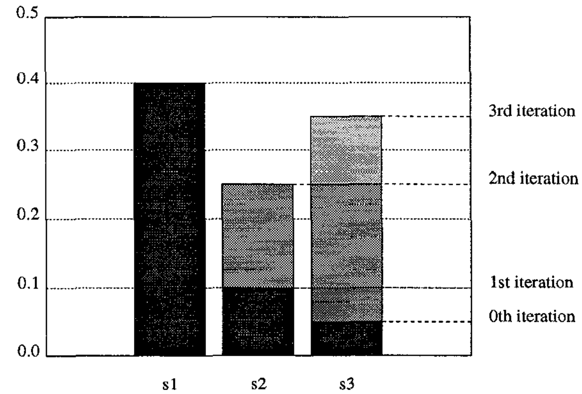

Figure 2: Graphical display of rate allocation for each session at each iteration in the peer-to-peer example.

- 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) 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. □

Remark 1 As can be noted in the above example, the GMM policy favors sessions with lower MCR requirements over those with higher MCRs in terms of sharing the remaining network capacity (link capacity minus sum of MCRs). This may appear to be (and in some circumstances is) "unfair" to sessions with high MCR requirements. We therefore advocate the use of the GMM policy only when network management policy is to discourage MCR-greedy users while being fair to users requesting the smallest MCR requirement for a given application. This situation may arise in networks where users are not explicitly charged for network resources used, e.g. most corporate enterprise networks. In such an environment, the GMM policy attempts to achieve equality in bandwidth sharing by first considering the session with the smallest MCR. □

Formally, the generalized max-min policy is defined as following:

Definition 2 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 3 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 $r_t > \mathrm{MCR}_t$. □

It can be shown that the following theorem holds for the GMM policy [5].

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. □

In Example 1, Link 12 is a GMM-bottleneck link for both $s1$ and $s3$ (see Definition 3). On the other hand, $s1$ and $s3$ have different rate allocations (0.4 for $s1$ and 0.35 for $s3$). Therefore, it is necessary 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 4 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 ℓ. Then τ_ℓ satisfies

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

where

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

Remark 2 In the special case when $\mathrm{MCR}^s = 0$ for every $s \in \mathcal{S}$, the GMM-bottleneck link rate τ_ℓ in Definition 4 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 in $\mathcal{U}_\ell$. This is exactly the expression for the max-min bottleneck link rate at link ℓ [2]. □

With the above clarification, it is clear that the GMM-bottleneck link rate at Link 12 is 0.35 in Example 1.

The centralized algorithm for the GMM policy requires global information and is therefore difficult to maintain in real world networks. To achieve the GMM policy in a distributed network environment, we need to design a distributed algorithm using only local information and achieving the GMM policy through distributed and asynchronous iterations.

3 An ABR Implementation for the GMM Policy

Our distributed implementing the GMM policy used the flow control framework for the ATM ABR service [1]. A generic rate-based closed-loop congestion control mechanism for ABR service is shown in Fig. 3. Resource Management (RM) cells are inserted periodically among ATM data cells to convey network congestion and available bandwidth information to the source. RM cells contain important information such as the source's allowed cell rate (ACR) (called the current cell rate (CCR) in the RM cell's field), minimum cell rate (MCR) requirement, explicit rate (ER), congestion indication (CI) bit and no increase (NI) bit.

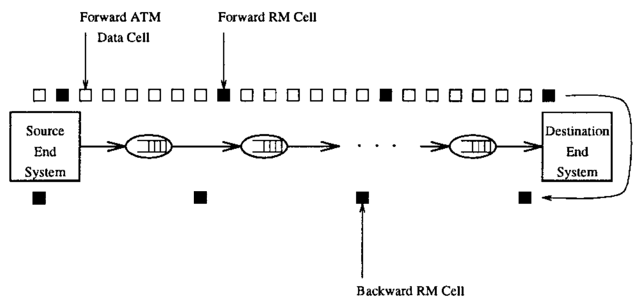

Figure 3: Rate-based closed-loop flow control for an ABR virtual connection.

Our ABR switch algorithm for GMM policy is based on the *Intelligent Marking* technique, originally proposed in [9] and further refined in [10, 11]. The key idea of this technique is to employ several variables at each output port of a switch to estimate the max-min bottleneck link rate. Using the ABR closed-loop feedback mechanism, the ER field of a returning RM cell is set to the minimum of all the estimated bottleneck link rates on all its traversing links, resulting in max-min rate allocation.

Fig. 4 illustrates the switch behavior of the Intelligent Marking technique [10, 11]. Four variables MCCR (Mean Current Cell Rate), UCR (Upper Cell Rate), EBR (Estimated Bottleneck Rate) and LOAD are defined for the following purpose:

MCCR Contains an estimated average cell rate of all VCs traversing this link;

UCR Contains an estimated upper limit of the cell rates of VCs traversing this link;

EBR Contains an estimated bottleneck link rate;

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 at each link, where the value of TLR is the target load ratio, and $0 < \alpha < 1$.

Algorithm 2 Intelligent Marking

Upon the receipt of RM(CCR, ER) from the source of

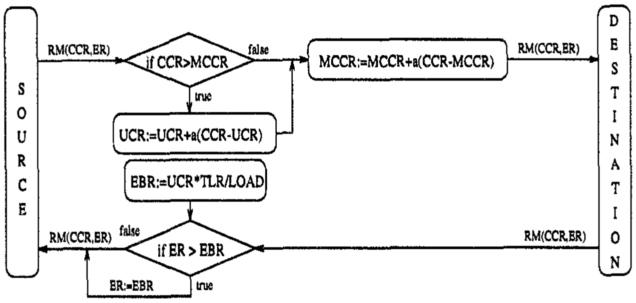

Figure 4: Switch behavior of Intelligent Marking protocol.

a VC:
    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 VC:
    EBR := UCR * TLR / LOAD;
    if (QS > QT), then
        EBR := (QT / QS) * EBR;[2]
    if (ER > EBR), then
        ER := EBR;
        Forward RM(CCR, ER) to its source;
    else
        Forward RM(CCR, ER) to its source. □

The Intelligent Marking technique (Algorithm 2) is a heuristic algorithm. We can only give an intuitive explanation on how it works. The RM cells from all VCs participate in the exponential averaging of MCCR with MCCR := MCCR + α(CCR − MCCR) while only some VCs with CCR greater than MCCR (potentially VCs 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 max-min bottleneck rate at a link and it is greater than or equal to any of the VC'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 for each VC.

The most attractive feature of the Intelligent Marking technique is its low implementation cost. It does not require each link of a switch to keep track of each traversing VC's state information (so called per-VC accounting) 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 policy. We will now extend the Intelligent Marking technique for the GMM policy.

Comparing the definitions for the max-min policy [2] and GMM policy (Definition 2), we observe that they are almost identical except the additional requirement in the GMM policy that a rate vector must be ABR feasible (see Definition 1). This motivates us to take the following steps to design an ABR algorithm for the GMM policy based on the Intelligent Marking technique.

1. Continue to use Intelligent Marking (Algorithm 2) as the switch algorithm for GMM policy. This will also satisfy the second requirement for ABR-feasibility ($F_\ell \le C_\ell$ for all $\ell \in \mathcal{L}$) due to the self-stabilizing nature of the Intelligent Marking technique, i.e. queue size is always kept finite.
2. Let each ABR source enforce the first requirement of ABR-feasibility, i.e. $\mathrm{MCR}_s \le r_s \le \mathrm{PCR}_s$ for all $s \in \mathcal{S}$.

The following algorithm specifies the source behavior of our ABR algorithm, which conforms to the framework of source behavior in [1].

Algorithm 3   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:
    ACR := max{min{(ACR + AIR), ER}, MCR}. □

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

4 Simulation Results

In this section, we implement our ABR switch algorithm on our network simulator [4] and perform simulations to demonstrate its effectiveness in achieving the GMM policy.

The ATM 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 VCs going through that port. At each output port of an ATM switch, we implement our ABR algorithm for the GMM policy.

The network configurations that we use are the peer-to-peer (Fig. 1) and the *parking-lot* (Fig. 5) network configurations.

Table 2 lists the parameters used in our simulation. The distance from source/destination to the switch is 100 m and the link distance between ATM switches is 10 km (corresponds to a LAN).

The Peer-to-Peer Network Configuration

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

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

End System	PCR	PCR
	MCR	MCR
	ICR	MCR
	Nrm	32
	AIR	3.39 Mbps
Link	Speed	150 Mbps
Switch	Cell Switching Delay	4 μSec
	α	0.125
	Load/Utilization Measurement Interval	500 μSec
	Queue Threshold for ER Adjustment	50 cells
	Output Buffer Size	2000 cells

Table 2: Simulation parameters.

Fig. 6 shows the ACR at source for sessions s1, s2, and s3, respectively with the MCR/PCR requirements for each session 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 VC matches the rates listed in Table 1. To study the network utilization of our ABR algorithm, we also show the inter-switch link utilization (Link 12) and the queue size of the congested switch (SW1) in Fig. 7. We find that the link is 100% utilized with reasonably small buffer requirements.

The Parking-Lot Network Configuration

The parking-lot configuration that we use is shown in Fig. 5 where sessions $s1$ and $s2$ start from the first switch and go to the last switch [7]. Sessions $s3$ and $s4$ start from SW2 and SW3, respectively, and terminate at the last switch. Here, Link 34 is the only potential GMM-bottleneck link.

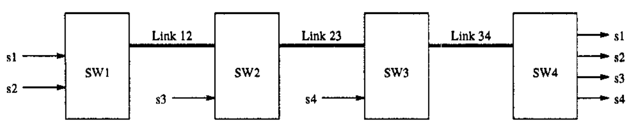

Figure 5: The parking lot network configuration.

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

Table 3: MCR requirement, PCR constraint, and GMM rate allocation of each session for the parking-lot network configuration.

Table 3 lists the MCR and PCR constraints for each session and the rate assignment for each session under the centralized GMM rate allocation algorithm.

Fig. 8 shows the normalized cell rates of each VC session under our distributed ABR algorithm. We see that they match fairly well with the rates listed in Table 3 after the initial transient period. Fig. 9 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 ABR algorithm achieves the GMM policy in a LAN environment.

5 Concluding Remarks

The main contributions of this work are the generalization of the theory of the max-min policy to include the minimum rate and peak rate constraints for each connection, and the development of a simple distributed algorithm consistent with the ATM Forum ABR traffic management framework to achieve the generalized max-min policy. Simulation results based on benchmark network configurations used by the ATM Forum demonstrated the effectiveness of our ABR algorithm in a LAN environment.

In a wide are network (WAN), the effectiveness of our heuristic ABR algorithm depends on careful system parameter tuning to minimize oscillations. Here, a more sophisticated ABR algorithm using per-VC accounting [5] may be necessary. But in a LAN environment, where implementation cost may well be the deciding factor in choosing an ABR algorithm, our simple algorithm offers satisfactory performance with minimal implementation cost ($O(1)$ storage requirements and computational complexity).

References

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

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

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

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

[5] Y. T. Hou, H. Tzeng, and 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.

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

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

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

[9] 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.

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

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

[12] N. Yin, "Max-Min Fairness vs. MCR Guarantee on Bandwidth Allocation for ABR," *Proc. IEEE ATM'96 Workshop*, San Francisco, CA, August 25-27, 1996.

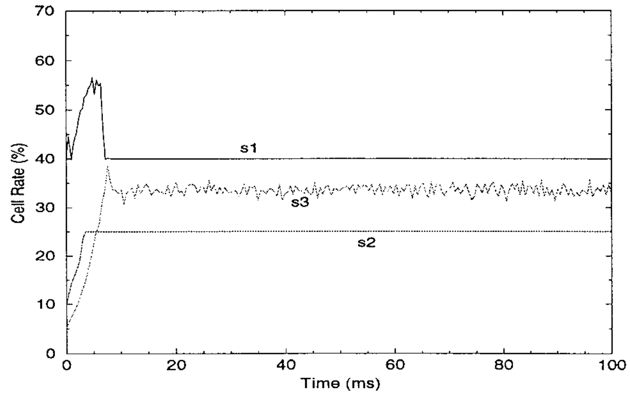

Figure 6: The cell rates of all connections in the peer-to-peer network configuration.

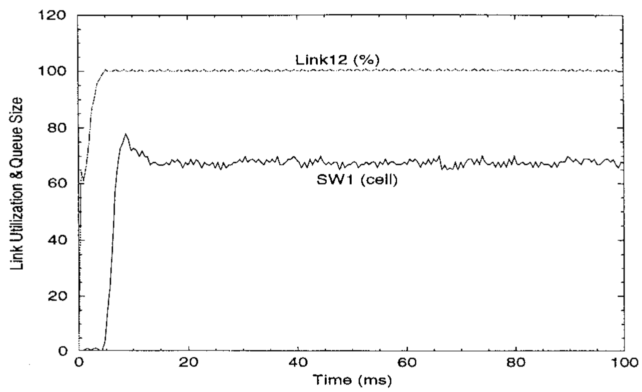

Figure 7: The link utilization and queue size of the congested switch in the peer-to-peer network configuration.

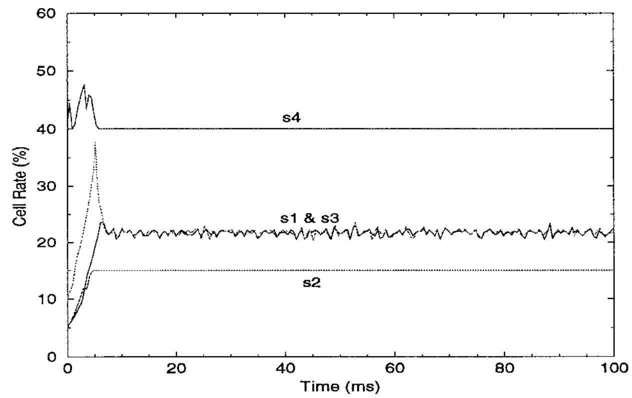

Figure 8: The cell rates of all connections for the parking-lot network configuration.

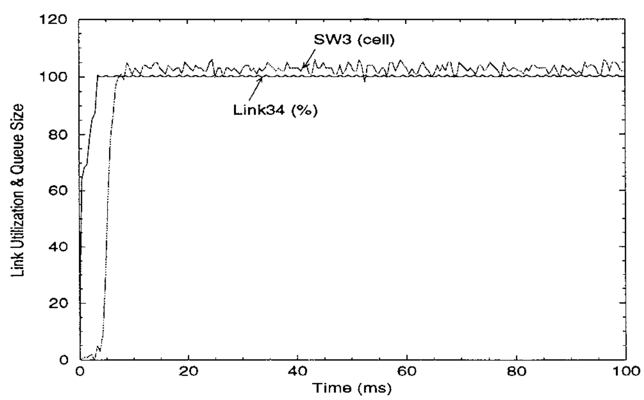

Figure 9: The link utilization and queue size of the congested switch for the parking-lot network configuration.