An Intelligent Explicit Rate Control Algorithm for ABR Service in ATM Networks

Ambalavanar Arulambalam, Xiaoqiang Chen

Bell Laboratories
Lucent Technologies
Murray Hill. New Jersey, USA.
email: arul@lucent.com, xchen@lucent.com

Nirwan Ansari

Dept. of Electrical & Computer Engineering
New Jersey Institute of Technology
Newark, New Jersey, USA
email: ang@njit.edu

ABSTRACT

The central issue of explicit rate control for Available Bit Rate (ABR) service in ATM networks is the computation of fair rate for every connection. In this paper, we propose a new fair-rate allocation algorithm called Fast Max-Min Rate Allocation *(FMMRA) for ATM switches supporting ABR service. The FMMRA algorithm provides the means to compute the max-min fair rates with $O(1)$ computational complexity. This exact calculation of fair rates expedites quick convergence to max-min fair shares, and offers excellent transient response. At the steady state, the algorithm operates without causing any oscillations in rates. The FMMRA algorithm does not require any parameter tuning and proves to be very robust in a large ATM network. Some simulation results are provided to show the effectiveness of the algorithm.*

1. INTRODUCTION

Available bit rate (ABR) service defined in the standard bodies is intended for applications with bursty traffic that are sensitive to cell loss, but can tolerate certain delay. It is also very difficult to predict bandwidth requirements of these applications. ABR is designed in such a way that these applications can grab any unused network resources (bandwidth and buffer space). Gains due to statistical resource utilization, however, come at the risk of potential congestion when many applications compete for the network resources. Proper congestion control must be in place to ensure that the network resources can be shared in a fair manner and that performance objectives such as cell loss ratio can be maintained. A flow control mechanism is specified by the ATM Forum in [1] which supports several types of feedback to control the source rate in response to changing transfer characteristics. This feedback information is conveyed to the source which adapts its traffic in accordance with the feedback. The feedback information includes the state of congestion and a fair share of the available bandwidth according to a network specific allocation policy.

A network switch is responsible for allocating the fair share of the bandwidth among all connections that compete at this switch point. Since the allocation policy is implementation-specific, it has been at the center of switch design and implementation for the last few years. This issue has been becoming one of the important differentiate factors for the next generation of commercially available switches. A number algorithms that calculate fair share have been proposed and studied in the literature, and they can be in general classified broadly into two approaches: fair rate approximation and exact fair rate computation, depending on the congestion monitoring criteria and congestion detection methods. Readers are referred to [2] for a survey of these algorithms.

The work presented here simplifies the algorithm presented in [3]. The work in [3] was an early proposal to compute explicit rate in a distributed manner, and originally formulated in the context of packet switching. The development of this algorithm has had a significant influence on the fair rate allocation algorithms for the ABR service. At the time of its development, much of the ABR specification did not exist. Thus, many of the features available now in RM cells were not utilized in its design. In this paper, we present an algorithm called *Fast Max-Min Rate Allocation* (FMMRA) algorithm, that calculates the max-min fair rates with $O(1)$ computational complexity.

The remainder of the paper is organized as follows. Section 2 summarizes the rate-based control framework. A detailed description of the FMMRA algorithm is presented in section 3. A simple network topology is simulated, and the results are presented in section 4. We conclude the paper by providing a summary and some remarks in section 5.

2. ABR CONGESTION CONTROL FRAMEWORK

The ABR congestion control scheme is a rate-based, closed-loop and per-connection control which utilizes the feedback information from the network to regulate the rate of transmitting cells at the source. The source generates special probe cells called *resource management* (RM) cells in proportion to its current data cell rate. The destination will turn around and send back the RM cells to the source in the backward direction. The RM cells which can be examined and modified by the switches in both forward and backward directions carry the feedback information of the state of congestion and the fair rate allocation. Details of the RM cell format can be found in [1]. A general end-to-end mechanism is depicted in Figure 1.

A switch shall implement at least one of the following methods to control congestion at queuing points: (a) *Explicit*

0-7803-3925-8/97 $10.00 ©1997 IEEE

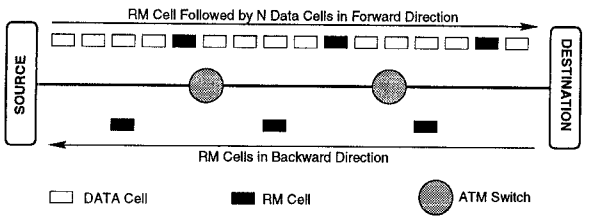


Figure 1: End-to-End Congestion Control for ABR Service

forward congestion indication (EFCI) marking, (b) Relative rate marking, (c) *Explicit rate* (ER) marking. Switches that implement options (a) and (b) are known as binary switches which can reduce implementation complexity but may result in unfairness, congestion oscillation and slow congestion response. The switches that implement option (c) are generally called ER switches which require sophisticated mechanisms in place at the switches for the computation of a fair share of the bandwidth. The standard-defined source and destination behaviors, however, allow the inter-operation of the above three options. The operational details are beyond the scope of this paper, and can be found in [1], [4].

3. THE FMMRA ALGORITHM

The most critical component of the fair rate allocation is to define a fair rate allocation policy. No set of connections should be arbitrarily discriminated against and no set of connections should be arbitrarily favored, although resources may be allocated according to a defined policy. A number of fairness polices are possible. A commonly used fairness criterion is *max-min fairness* which was introduced in [5].

At any given link in the network, connections that are competing for bandwidth can be grouped into two categories:

- Bottlenecked connections: those connections that are unable to achieve their fair (equal) share of bandwidth at the given link because of constraints imposed by their PCR requirements or most likely by limited bandwidth available at other links;
- Non-bottlenecked connections: those connections for which their achievable bandwidths are only limited at the given link (usually referred to as a bottleneck link).

The goal of the FMMRA algorithm is to find the maximum rate that a link, l, can allocate for a given connection. This rate is referred to as the *advertised rate*, γ_l. If a connection cannot use the advertised rate, the connection is marked as a bottlenecked elsewhere, and its bottleneck bandwidth is recorded. This implies that there is additional bandwidth available that can be shared by other connections. The advertised rate is recomputed incorporating the bottleneck status of the connection. At steady state the advertised rate reflects the maximum allocated rate on the link.

3.1. Updating Rule

The objective of any max-min fair allocation algorithm is to identify the bottleneck bandwidth of each connection in an iterative manner. This is done by first maximizing the link capacity that is allocated to the connections with the minimum allocation, and then using the remaining link capacity for other connections, in a way that it maximizes the allocation of the most poorly treated among those connections. For the purpose of determining the fair allocation rates for other connections, the rate of the bottleneck connections will be fixed, and a reduced network, where the bottlenecked connections are eliminated and the capacity of all the links (excluding bottleneck links) is reduced by the total bottleneck bandwidth elsewhere, is considered. The procedure is repeated in the reduced network until all the connections have received the fair allocation rates. We could accomplish the above task by keeping track of each connection's bottleneck status and computing the advertised rate, γ_l, as follows:

$$\gamma_l = \frac{C_l^A - \bar{C}_l}{N_l - \bar{N}_l}, \tag{1}$$

where C_l^A is the available bandwidth for ABR traffic, $\bar{C}_l$ is the sum of the bandwidth of connections bottlenecked elsewhere, N_l is the total number of connections traversing link l, and $\bar{N}_l$ is the total number of bottlenecked connections elsewhere.

The advertised rate, γ_l, is updated every time a backward RM cell is received at the link. Let t be the time when a backward RM cell is received, and let t^+ be the time of the new update of the advertised rate based on information from the recently received RM cell. Denote $\gamma_l(t)$ as the advertised rate when a backward RM cell is received and $\gamma_l(t^+)$ as the new advertised rate results after the update. By incorporating time indices in Equation (1) we get

$$\gamma_l(t^+) = \frac{C_l^A - \bar{C}_l(t^+)}{N_l - \bar{N}_l(t^+)}. \tag{2}$$

Note that in Equation (2), $\bar{C}_l(t^+)$ is the sum of the total bottleneck bandwidth prior to the update, $\bar{C}_l(t)$, and change in the bottleneck bandwidth of the connection, $\Delta\lambda$, at the time of update. Similarly, $\bar{N}_l(t^+) = \bar{N}_l(t) + \Delta\beta$, where $\Delta\beta$ represents the change in bottleneck status of the connection. The bottleneck status and bottleneck bandwidth of a connection is determined by comparing the ER field in the RM cell and the advertised rate of the link. Let β_l^i be an indicator to decide if the connection, i, is bottlenecked elsewhere, and λ_l^i be the corresponding bottleneck bandwidth. Based on the explicit rate value, λ_i^{ER}, found in the received RM cell, the variables $\Delta\lambda$ and $\Delta\beta$ are computed as follows:

$$\Delta\lambda = \begin{cases} \lambda_i^{ER} - \lambda_l^i\beta_l^i & \text{if } \lambda_i^{ER} < \gamma_l, \\ -\lambda_l^i\beta_l^i & \text{if } \lambda_i^{ER} \geq \gamma_l, \end{cases} \tag{3}$$

$$\Delta\beta = \begin{cases} 1 - \beta_l^i & \text{if } \lambda_i^{ER} < \gamma_l, \\ -\beta_l^i & \text{if } \lambda_i^{ER} \geq \gamma_l. \end{cases} \tag{4}$$

Now Equation (2) becomes

$$\gamma_l(t^+) = \frac{C_l^A - \bar{C}_l(t) - \Delta\lambda}{N_l - \bar{N}_l(t) - \Delta\beta}. \quad (5)$$

Equation (5) can be written as

$$\gamma_l(t^+) = \frac{C_l^A - \bar{C}_l(t)}{N_l - \bar{N}_l(t)} + \frac{\frac{C_l^A - \bar{C}_l(t)}{N_l - \bar{N}_l(t)}\Delta\beta - \Delta\lambda}{N_l - \bar{N}_l(t) - \Delta\beta}. \quad (6)$$

Note that in Equation (6)

$$\frac{C_l^A - \bar{C}_l(t)}{N_l - \bar{N}_l(t)} = \gamma_l(t), \quad (7)$$

where $\gamma_l(t)$ denotes the advertised rate just before the update. The new advertised rate, $\gamma_l(t^+)$, can be expressed in terms of the old advertised rate, $\gamma_l(t)$:

$$\gamma_l(t^+) = \gamma_l(t) + \frac{\gamma_l(t)\Delta\beta - \Delta\lambda}{N_l - [\bar{N}_l(t) + \Delta\beta]}. \quad (8)$$

3.2. ABR Connection Management

For the FMMRA algorithm, two per-connection variables are used to keep track of status of the connection. A one bit variable, β_l^i, will be used to decide if the connection, i, is bottlenecked elsewhere, and the corresponding bottleneck bandwidth is recorded in variable λ_l^i. These variables are referred and updated upon arrival of RM cells.

In addition to the connection table, each queue maintains variables such as the total number of ABR connections using the queue (N_l), the advertised rate (γ_l), the total number of ABR connections bottlenecked elsewhere ($\bar{N}_l$), and available bandwidth information. When a connection is opened or closed, the number of ABR connections, N_l, is updated as follows:

$$N_l = \begin{cases} N_l + 1 & \text{if a new VC opens at link } l, \\ N_l - 1 & \text{if an existing VC closes at link } l. \end{cases} \quad (9)$$

At the time when the connection opens, the per-connection variables are initialized to zero, and the advertised rate is reduced as follows:

$$\gamma_l = \gamma_l - \frac{\gamma_l}{N_l - \bar{N}_l}. \quad (10)$$

When a connection closes, the information regarding this connection should be erased, and the status of the link should be adjusted accordingly. Let j be the connection that was closed. The updating is as follows:

$$\bar{N}_l = \bar{N}_l - \beta_l^j, \quad (11)$$

$$\gamma_l = \gamma_l + \frac{\lambda_l^j + \gamma_l\beta_l^j}{N_l - \bar{N}_l}. \quad (12)$$

3.3. Explicit Rate Calculation

The rate-based approach specifies that a switch should not increase the ER field but could reduce the field to a lower value. The algorithm achieves this by comparing the ER field in the RM cell with the advertised rate, and rewriting the ER field as follows:

$$\lambda_i^{ER} \leftarrow \min(\gamma_l, \lambda_i^{ER}). \quad (13)$$

The above assignment is done whenever an RM cell is received at the switch, regardless of the direction of the RM cell. The downstream switches learn the bottleneck status of each connection whenever a forward RM cell is marked by an upstream switch, and the upstream switches learn the bottleneck status of a connection whenever a backward RM cell is marked by a downstream switch. This bi-directional updating of ER in the RM cell is a key feature of FMMRA that plays a significant role in drastically reducing the convergence time of max-min fair rate allocation process. The advertised rate is updated only when a backward RM cell is received, since this RM cell has been seen by all the switches along its path, and the fields contain the most complete information about the status of the network.

At the reception of a backward RM cell, the change of status of the connection is determined by calculating the change in per-VC variables, $\Delta\lambda$ and $\Delta\beta$, using Equation (4), and the new advertised rate is calculated as follows:

$$\gamma_l = \begin{cases} C_l^A & \text{if } N_l = 0, \\ \gamma_l + \frac{\gamma_l\Delta\beta - \Delta\lambda}{N_l - [\bar{N}_l + \Delta\beta]} & \text{if } N_l > \bar{N}_l, \\ \gamma_l & \text{if } N_l = \bar{N}_l. \end{cases} \quad (14)$$

In contrary to the algorithm presented in [3], which requires the switch inspecting the state of all the connections, the FMMRA algorithm only requires the knowledge of the connection which is seen by the switch at the time of update. This feature makes the computational complexity of FMMRA to be of $O(1)$, whereas the algorithm in [3] has a computational complexity of $O(N_l)$.

Next, the new number of bottlenecked connections is updated as follows:

$$\bar{N}_l = \begin{cases} \bar{N}_l + 1 - \beta_l^i & \text{if } \lambda_i^{ER} < \gamma_l, \\ \bar{N}_l - \beta_l^i & \text{if } \lambda_i^{ER} \geq \gamma_l. \end{cases} \quad (15)$$

Finally, once the explicit rates are marked on any backward RM cell and the port variables are updated, the switch updates the corresponding per-connection variables in the connection table as follows:

$$\beta_l^i = \begin{cases} 1 & \text{if } \lambda_i^{ER} < \gamma_l, \\ 0 & \text{if } \lambda_i^{ER} \geq \gamma_l, \end{cases} \quad (16)$$

$$\lambda_l^i = \begin{cases} \lambda_i^{ER} & \text{if } \lambda_i^{ER} < \gamma_l, \\ 0 & \text{if } \lambda_i^{ER} \geq \gamma_l. \end{cases} \quad (17)$$

It can be shown that this method of fair rate calculation will converge and the convergence time is finite. Due to space constraints, the proof of convergence which can be found in [6], is not presented here. The convergence time is shown

to be proportional to the number of distinct fair allocation rates and the round trip time delays. In order to give an approximate value of an upperbound, let us assume that the worst case round trip time delay is D, and let M be the distinct fair allocation rates. It can be shown that the algorithm will converge to the fair allocation rates approximately within MD time units.

3.4. Enhancements

In the above approach, note that only the ER value in the RM cell is used to calculate fair share in both directions. It can be argued that if a switch along a connection's path does not use any ER approach, the algorithm will fail, since it relies completely on ER values in the RM cell. Moreover, this approach is conservative due to the fact that if any connection becomes idle, the unused bandwidth will not be reallocated, and the link will be under utilized. In order to overcome these limitations some enhancements to the basic algorithm are necessary.

In order to overcome these limitations, it is possible to incorporate the traffic load measurements and queue length information into ER calculations. Similar to the popular ERICA algorithm [7], the ABR load factor, ρ_l, is used to indicate the level of congestion. The load factor is the ratio of actual ABR traffic bandwidth, $F_l(t)$, and the available bandwidth for ABR traffic, $C_l(t)$. $F_l(t)$ is equivalent to the sum of current cell rates of all the ABR connections or total flow on the link. The load factor can be computed as

$$\rho_l(t) = \frac{F_l(t)}{C_l^A(t)}. \tag{18}$$

The load factor reflects how well the ABR bandwidth is utilized, for example, $\rho_l < 1$ reflects that some of the connections are sending cells at a rate less than their allowed rate. This presents an opportunity for the non-bottlenecked connections in link l to increase their rate. This is done by modifying Equation (13) as follows:

$$\lambda_i^{ER} \leftarrow \min\left\{\max\left(\frac{\hat{\lambda}_l^i(1-\beta_l^i)}{\rho_l}, \gamma_l\right), \lambda_i^{ER}\right\}, \tag{19}$$

where $\hat{\lambda}_l^i$ could denote the *current cell rate* (CCR) value for connection i at link l at the time of estimation of ρ_l. A future contribution will address the possibilities of using an alternative reference rate instead of CCR. In addition to load factor, queue length thresholds are utilized to control queue lengths effectively. If the queue length reaches a threshold (Q_T), the operation as in Equation (19) is turned off, and the assignment as in Equation (13) is turned on. This ensures that whenever queue length is above a threshold (i.e., congested), even if some connections are idle, the non-idle connections are not given any additional bandwidth. Furthermore, if the queue length is above a high threshold, DQT (i.e., highly congested), the available bandwidth for ABR is reduced as a function of the queue length, to allow the queue to drain and operate at a desired queue level.

4. SIMULATION RESULTS

A network topology, shown in Figure 2, is simulated. For performance measurement purposes, the switches are assumed to be non-blocking and output-buffered. The sources are assumed to be well behaved, persistently greedy, and always transmit at the maximum allowed cell rate. We use this network model to study the algorithm's performance in Local Area Network (LAN) and Wide Area Network (WAN) configurations. In a LAN configuration, all the links are $1Km$ in length. In a WAN configuration, the distances of links L1 and L12 are $1000Km$ and $100Km$, respectively. In both cases, all the links have 100 Mbps capacity and a propagation delay of $5\mu s$ per Km.

For both cases we consider the situation where some connections are bottlenecked. In our simulations, we make connections 1 and 3 bottlenecked by setting the *peak cell rate* (PCR) value of the sources S1 and S3 to 5 Mbps. The sources S2, S4, and S5 are given a PCR value of 150 Mbps. This implies that connection 2 can use 90 Mbps on link L12 until $t = 100ms$. At any time $t > 100ms$, connections 2, 4 and 5 should receive equal share of bandwidth on link L12. Thus, the steady state bandwidth share for connections 1 and 3 is 5 Mbps, and 30 Mbps for connections 2,4 and 5. The instantaneous bandwidth of each connection and the total link utilization are plotted in Figure 3.

An algorithm that uses $\frac{C_l^A}{N_l}$, instead of max-min rate equation is ERICA. In order to illustrate a possible problem with this approach the network is simulated using ERICA. From Figures 3(a) and 3(b), it can be seen that the ERICA algorithm fails to converge to the fair rates, for both LAN and WAN environments. The connections that start late do not get their fair share of 30Mbps; instead they get a share of 20Mbps. The FMMRA algorithm results in oscillation free rates at steady state, as shown in Figures 3(c) and 3(d).

5. SUMMARY AND CONCLUSIONS

In this paper we have presented and evaluated a fair, fast adaptive rate allocation algorithm designed for ATM switches implementing explicit rate congestion control supporting ABR services. The most important features of this algorithm are $O(1)$ computational simplicity, fast convergence, oscillation free steady state, high link utilization and low buffer requirements. The algorithm takes advantage of per-connection accounting, to enable the switches to calculate an exact fair rate, and to provide the means for quick convergence. The use of advertised rate as a common reference point to all the connections enable the switches to quickly allocate max-min fair rates. Another advantage is that the algorithm uses only ER as feedback. The algorithms that use binary mechanisms (i.e., CI or NI as feedback) often exhibit parameter dependency. In particular, the value of *additive increase factor* (RIF) is needed to be small. Moreover, improper selection of RIF often leads to unfairness. It is observed that the FMMRA algorithm converges to the correct fair rates without any oscillations regardless of the RIF used. In fact, using

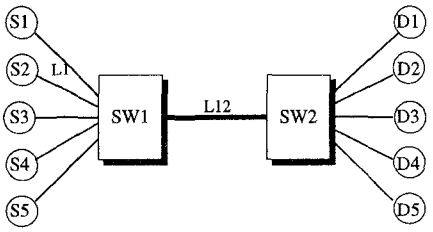


Figure 2: Network Model

a very high RIF value results in fast convergence.

6. REFERENCES

[1] The ATM Forum, "The ATM Forum Traffic Management Specification,Version 4.0," *ATM Forum Contribution, AF-TM 96-0056.000*, April 1996.

[2] A. Arulambalam, X. Chen and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM Networks," *IEEE Communications Magazine*, vol. 34, pp. 92–100, November 1996.

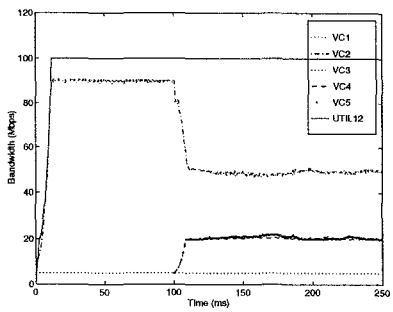
[3] A. Charny, D.D. Clark and R. Jain, "Congestion Control with Explicit Rate Indication," *Proc. ICC 95*, June 1995.

[4] F. Bonomi and K. W. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service," *IEEE Networks Magazine*, pp. 25–39, March-April 1995.

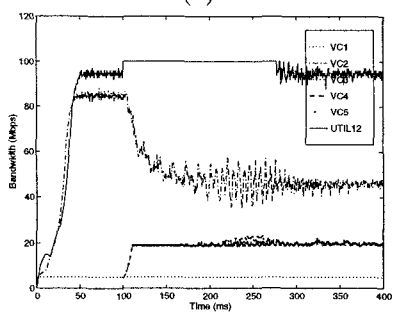
[5] J. M. Jaffe, "Bottleneck Flow Control," *IEEE Transactions on Communications*, vol. 29, pp. 954–962, July 1981.

[6] A. Arulambalam, X. Chen and N. Ansari, "A New Fair-Rate Allocation Algorithm for Available Bit Rate Service in ATM Networks," Tech. Rep. CCSPR-96-NA2, Electrical and Computer Engineering, New Jersey Institute of Technology, February 1996.

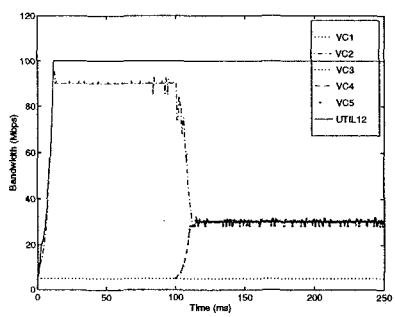
[7] R. Jain, S. Kalyanaraman and R. Viswanthan, "A Sample Switch Algorithm," *The ATM Forum Contribution AF-TM 95-0178*, February 1995.



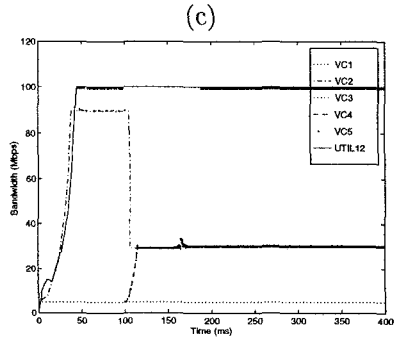
(a)



(b)



(c)



(d)

Figure 3: Instantaneous Bandwidth Utilization in Single-Hop Bottleneck Configuration: (a) ERICA algorithm (LAN) (b) ERICA algorithm (WAN) (c) FMMRA algorithm (LAN) (d) FMMRA algorithm (WAN) - For all cases: $N_{RM} = 32$, ICR=2Mbps, MCR = 0, RIF=1/8