

A NOVEL EFFICIENT AND ROBUST EXPLICIT RATE CONGESTION AVOIDANCE ALGORITHM FOR ABR SERVICE IN ATM NETWORKS

Tanun Jaruvitayakovit, Naris Rangsinoppamas,
Boonchoung Tansuthepverawongse and Prasit Prapinmongkolkarn
*Telecommunications System Research Laboratory, Department of Electrical Engineering,
Faculty of Engineering, Chulalongkorn University, Phaya Thai Road, Bangkok 10330, Thailand*

Key words: ABR service, Explicit Rate (ER) calculation, Max-min fairness

Abstract: This paper investigates the problems in explicit rate congestion avoidance algorithms for ABR service in ATM networks. It is found that the performance of the existing algorithms [2, 3, 4] depend on network conditions which vary dramatically in the real condition such as number of traversing sessions, traffic characteristics and propagation delay. In addition, this paper proposes a novel efficient and robust ABR congestion avoidance algorithm called “Fast Rate Allocation Congestion Avoidance (FRACA)”. Some simulation results are presented to illustrate performance in terms of computation accuracy and switch buffer occupancy of the proposed algorithm. Moreover, the simulation results are compared with Explicit Rate Indication Congestion Avoidance (ERICA+) and Enhanced Fast Max-Min Rate Allocation (E-FMMRA) which are the well-known explicit rate switch algorithms and also compatible with ATM Forum Traffic Management Specification 4.0. From comparison, the proposed algorithm works better than both algorithms for all configuration scenarios.

1. INTRODUCTION

Available Bit Rate (ABR) service in ATM networks was designed in standard body of Traffic Management [1] for supporting applications that tolerate to delay but sensitive to cell loss. ABR service category uses close-

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35581-8_35](https://doi.org/10.1007/978-0-387-35581-8_35)

loop feedback for flow control. Resource Management (RM) cell is sent to network by the sources after sending Nrm-1 user data cells to the network. The destination turns around and sends the RM cell in backward direction which is known as backward RM cell. The intermediate switches compute the proper rate and convey it in the Explicit Rate (ER) field in RM cell. After receiving backward RM cell, all sources adjust their traffic rate in accordance with the feedback information in backward RM cell.

The main goals for switch algorithm design are the share of free bandwidth (left over from the higher priority traffic such as CBR and VBR services) to all sources in a fair manner, maximize network utilization with controllable switch queue length. Another important topic for the ABR service is that the above goals should be achieved regardless of network conditions such as number of traversing sessions, traffic characteristics and system propagation delay. The contribution of this paper is to develop a novel, effective and robust algorithm which can achieve the goals regardless of network conditions unlike the previously proposed algorithms [2, 3, 4] which parameters setting mainly depend on network conditions.

This paper is organized as follows. Section II describes and discusses the problems associated with existing algorithms (ERICA+ and E-FMMRA). Section III describes the proposed algorithm and points out the advantages of the proposed algorithm. In section IV, we used simulation results to illustrate performance of the proposed algorithm compared with ERICA+ and E-FMMRA algorithm. Finally, the conclusion is given in section V.

2. MOTIVATION AND RELATED WORK

ERICA+ [4] and E-FMMRA [2, 3] are the well-known explicit rate ABR congestion control algorithms in ATM networks. Both algorithms are exact fair rate calculation algorithms [7].

Explicit Rate Indication Congestion Avoidance (ERICA+)

The algorithm calculates fair share rate which can be defined as

$$\text{FairShare} = \frac{\text{ABR Target Rate}}{\text{Number of active connections}} \quad (1)$$

where ABR target rate is the multiplication result of the left over bandwidth (from higher priority service class) and hyperbolic queue control function $[f(Q)]$ which defined as

$$f(Q) = \begin{cases} \frac{b \times Q_0}{(b-1) \times Q + Q_0} & \text{for } 0 \leq Q \leq Q_0 \\ \max\left(QDLF, \frac{a \times Q_0}{(a-1) \times Q + Q_0}\right) & \text{for } Q > Q_0 \end{cases} \quad (2)$$

where a and b are fixed parameters set to 1.15 and 1.0, respectively [4]. T_0 , which is converted into the target queue length (Q_0), specifies the target queuing delay. Queue Drain Limit Factor (QDLF) is the parameter to limit the rate of queue drain and set to 0.5. For queuing delays smaller than T_0 , the hyperbola is controlled by parameter b (called b -hyperbola). On the other hand, a -hyperbola determines how much drain capacity is used for draining out the queues built up. More drain capacity is allocated when the queue lengths are larger, up to a maximum of $(1 - QDLF)$.

The algorithm computes load factor (Z) based on ABR input rate and ABR target rate. Utilizing the load factor Z , the term VCShare is calculated as

$$VCShare = \frac{CCR}{Z} \quad (3)$$

where CCR (Current Cell Rate) is derived from the received forward RM cell to ensure that the most current information is used to provide fast feedback. In order to guarantee max-min fairness, the terms MaxAllocCurrent and MaxAllocPrevious are introduced. MaxAllocCurrent is the maximum value of every computed ER for all connections during current averaging interval and also MaxAllocPrevious which is the maximum value of every computed ER for all connections during previous averaging interval. Upon reception of a backward RM cell, the ER field is marked down as follows:

$$\begin{aligned} &\text{if } (Z > 1+\delta) \quad \dots \text{ where } \delta \text{ is a small value, typically set to } 0.1 \\ &\quad \text{ER in BRM cell} = \min(\text{ER in BRM cell}, \max(\text{fairShare}, VCShare)) \\ &\text{else } \text{ER in BRM cell} = \min(\text{ER in BRM cell}, \\ &\quad \max(\text{fairShare}, VCShare, \text{MaxAllocPrevious})) \end{aligned} \quad (4)$$

The performance of the ERICA+ algorithm depends significantly upon the way measurements are done and its parameters chosen [4]. ERICA+ algorithm suffers from achieving convergence in case of a large number of connections [6] as the effect from receiving all BRM cells during different averaging interval. In complex configuration, for example Generic Fairness Configuration 2 (GFC-2) [5], the convergence time is very long. In addition, at steady state the queue level cannot be conducted to the targeted value but it oscillates around the targeted value instead because the working region of the algorithm is in overload condition ($1 < Z < 1+\delta$). The queue control functions (step, linear, hyperbolic) mainly effect performance of the

algorithm [8]. Although with hyperbolic queue control function, the algorithm mostly works well but the computation complexity is very high compared to others.

Enhanced Fast Max-Min Rate Allocation (E-FMMRA)

The algorithm computes the rate it can support which is known as advertised rate (γ). If a connection cannot use the advertised rate, it is marked as bottlenecked elsewhere and its bottleneck bandwidth is recorded. This implies that there is additional bandwidth available that can be shared by other connections. The ER field in the received RM cell is read and marked in both direction to speed up the rate allocation process. The allocated rate to all non-bottleneck sessions is recorded as the maximum value of ER, denoted as ER_{\max} . The algorithm computes ER_{adjust} according to (5) every time the switch receives FRM and BRM cell.

$$ER_{\text{adjust}} = \frac{ER_{\max}}{\text{Load factor}} \quad (5)$$

Where Load factor is the ratio between ABR input rate and ABR target rate. The load factor reflects how well the ABR bandwidth is utilized. If switch queue length is lower than the high threshold (DQT), the algorithm updates ER_{\max} according to (6) when the switch receives BRM cell.

$$ER_{\max} = (1 - \alpha) \times ER_{\max} + \alpha \times \max(ER, ER_{\text{adjust}}) \quad (6)$$

α is an averaging factor and set at 1/8. For the heavy congestion condition (queue level is greater than the high threshold (DQT)), ER_{\max} is set to the maximum value between advertised rate and ER_{adjust} , that is

$$ER_{\max} = \max(\gamma, ER_{\text{adjust}}) \quad (7)$$

This means that non-bottlenecked sessions are not given any extra bandwidth (if available) in order to drain the queue to the targeted region. The algorithm updates the ER filed in a RM cell both in forward and backward direction for fast transient response [3]. The ER field in forward direction is updated according to

$$ER \text{ in FRM cell} = \min(ER \text{ in FRM cell}, \max(\gamma, (1 - \beta) \times ER_{\text{adjust}})) \quad (8)$$

where β is a single bit value indicating the session is bottleneck elsewhere. For backward direction, the allocated rate is updated as one of follows

$$ER \text{ in BRM cell} = \min(ER \text{ in BRM cell}, \max(\gamma, (1 - \beta) \times ER_{\text{adjust}})) \quad (9)$$

$$\text{or } ER \text{ in BRM cell} = \min(ER \text{ in BRM cell}, \max(\gamma, (1 - \beta) \times ER_{\max})) \quad (10)$$

$$\text{or } ER \text{ in BRM cell} = \min(ER \text{ in BRM cell}, \gamma) \quad (11)$$

The backward rate allocation (9-11) is employed upon the level of network congestion (switch queue length and load factor). If the switch is

not congest, queue level is lower than the set low threshold (QT), or switch is in moderate congestion condition, queue level locates between QT and DQT, and load factor is greater than one, the switch allocates ER according to (9). If switch is in moderate congestion condition and load factor is lower than or equal to unity, the switch computes ER regards (10). Finally if switch is in heavy congestion condition, the allocated rate is computed regards (11).

Actually sources traffic characteristic mainly effects performance of the algorithm. If all sources are persistent, the algorithm always converge to the max-min fairness with very fast transient response. In the case of source-bottleneck configuration, some sources send their traffic at the rate below the proper rate, the algorithm diverges from the targeted working point as the effect from setting too short averaging interval (as recommend in [3]) and updating forward and backward RM cell every time the switches receive. Moreover, we also found other two drawbacks of the algorithm. First, the algorithm suffers from achieving convergence in multi-link rate configuration as the effect of marking both forward and backward RM cell. Second, the algorithm cannot actually be forced to drain the switch queue level to the targeted region when all sessions are non-bottleneck sessions.

From robustness point of view, the performance of the congestion control algorithm should not depend on network condition, the more realistic case. In the other word, the algorithm should converge to the targeted working point with fast transient response regardless of network condition, for example number of connections, sources traffic characteristic and propagation delay.

3. THE FAST RATE ALLOCATION CONGESTION AVOIDANCE (FRACA) ALGORITHM

The main goals for developing the novel congestion control algorithm is convergence according to max-min fairness criterion with very fast transient response, 100% link utilization and controllable switch queue length with the simplest queue control function regardless of network conditions. The algorithm works only in backward direction and does not use CCR field in RM cell to compute the feedback rate. The connections which cannot use the advertised rate (A_{\max}) are marked as non-bottlenecked connection (at this switch) and their bandwidth is recorded. The rest bandwidth is equally shared among bottlenecked connections (at this switch). Marking only backward RM cell means we let only upstream switches to know status of downstream switches. However load factor is employed in order to increase network utilization in case there is some connections send their traffic at the rate below the proper rate and upstream bottleneck scenario. Moreover,

system parameters are updated once per connection per averaging interval which set long enough in order to accurately estimate traffic characteristic while maintaining fast transient response. From our study, setting averaging interval to 1500 cells works well for all scenarios (LAN, WAN and satellite ATM networks).

Basically, count-based setting of average interval (timer will be expired if the amount of ABR data cells reach the set threshold) is better than time-based setting of average interval especially if works with background VBR traffic. [6] illustrated that congestion control algorithm which uses per averaging computation and working region located in overload condition (ex. ERICA+) cannot work well with count-based averaging interval. So, the working region of the proposed algorithm is set at load factor equals unity (input rate equal served rate) and uses common weighted averaging for computation the feedback ER according to (6).

Non-bottlenecked sessions receive advertised rate as allocated ER. That is

$$ER \text{ in BRM cell} = \min (ER \text{ in BRM cell}, A_{\max}) \quad (12)$$

Otherwise, the bottlenecked sessions get ER_{\max} as feedback ER.

$$ER \text{ in BRM cell} = \min (ER \text{ in BRM cell}, ER_{\max} / Z) \quad (13)$$

Moreover, we use switch queue level as the additional parameter for computation the feedback ER. The proposed algorithm works well with step queue control function, the simplest queue control function [8]. The goal of queue control is keeping stable non-zero queue level at steady state. There are two important reasons for keeping stable non-zero queue level. First, queue control function is used to compensate for measurement and feedback errors caused by the system. Second, for configuration which has background VBR traffic, it is more reasonable to keep queue level at small level (non zero) because these queue will drain simultaneously in the case VBR traffic is in off phase (no need to wait for BRM cells that will take time to inform the sources) imply 100% link utilization. Our algorithm uses 2 levels which are low threshold (QT) and high threshold (DQT) for controlling queuing delay. The main objective is to keep switch queue length between QT (500 cells) and DQT (2000 cells). The algorithm computes $adjusted_ER$ which used to control queue level as follows

$$adjusted_ER = \frac{(f(Q)-1) \times ABR \text{ Capacity}}{N - N_{bottlenecked}} \quad (14)$$

where $f(Q)$ is the step queue control function shown in Table 1, N is number of established connections and $N_{bottlenecked}$ is number of bottlenecked connections.

Table 1. Step queue control function

Switch queue length (cells)	$f(Q)$
$0 \leq Q \leq 500$	1.02
$500 < Q \leq 2000$	1.00
$2000 < Q \leq 3000$	0.98
$3000 < Q \leq 4000$	0.95
$4000 < Q \leq 5000$	0.90
$5000 < Q \leq 6000$	0.80
$6000 < Q$	0.60

For the bottlenecked connections

$$\text{ER in BRM cell} = \text{ER in BRM cell} + \text{adjusted_ER} \quad (15)$$

The queue control function is also employed to compute load factor (Z) as follows

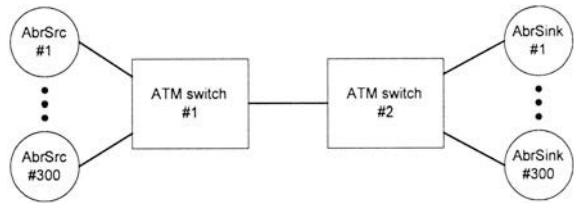
$$Z = \frac{\text{ABR Input Rate}}{\text{ABR Capacity} \times f(Q)} \quad (16)$$

4. SIMULATIONS AND DISCUSSIONS

We use 3 network configuration scenarios to illustrate performance of the proposed algorithm compared with ERICA+ and E-FMMRA algorithm. The first configuration “300 source configuration” is employed to present the problem regarding divergence of sources Allowed Cell Rate (ACR) in ERICA+ algorithm leading to growing unbound of switch queue length although the algorithm employs queue control function. The second configuration “Parking-lot + Link bottleneck + Source bottleneck Configuration” is done to show the problems associated with divergence of allocated rates by E-FMMRA algorithm in multi-link rate and parking-lot configuration and also long convergence time in ERICA+ algorithm. The third configuration scenario “Generic Fairness Configuration 2 (GFC-2)” is simulated to show the problems regarding the divergence of allocated rate in E-FMMRA algorithm, convergence time and buffer occupancy in ERICA+ algorithm.

Simulation results

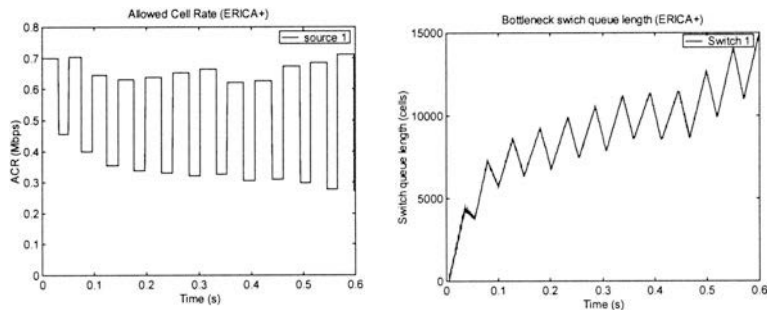
1. 300 source configuration



Parameters setting:

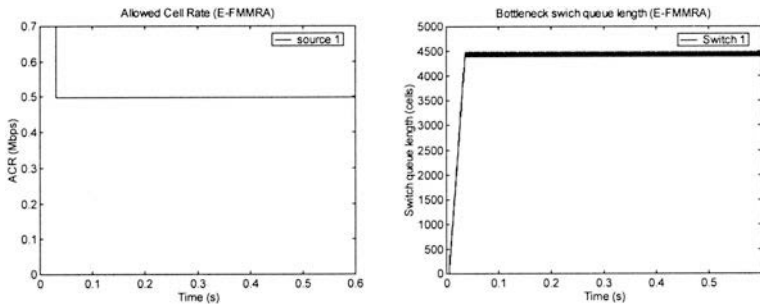
- All sources are persistent sources, PCR = Link capacity = 149.76 Mbps., MCR = 0.0 Mbps., ICR = 0.7 Mbps.
- All link propagation delay = 5 msec.

Figure 1. 300 source configuration



- (a) Allowed Cell Rate in Mbps. (b) Bottlenecked switch queue in cells

Figure 2. Results for 300 source configuration (ERICA+)



- (a) Allowed Cell Rate in Mbps. (b) Bottlenecked switch queue in cells

Figure 3. Results for 300 source configuration (E-FMMRA)

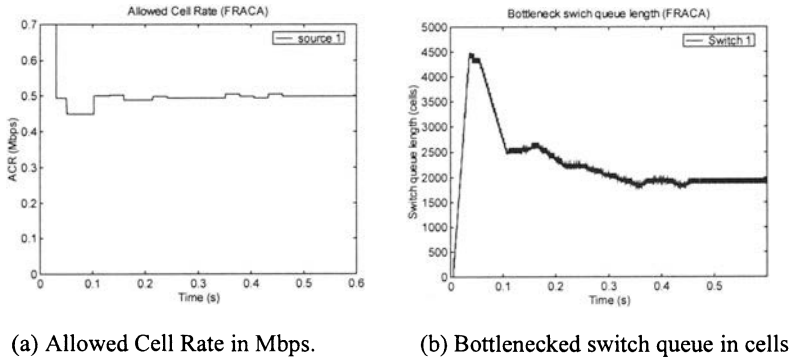
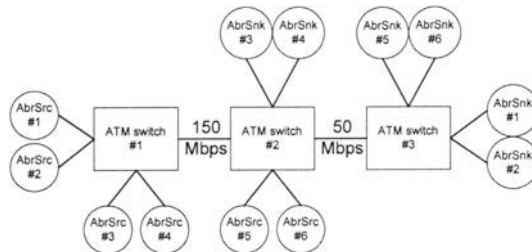


Figure 4. Results for 300 source configuration (FRACA)

This configuration is presented in order to illustrate effects of a large number of VCs to the congestion control algorithms. The expected rate for all connections is $149.76/300 \approx 0.5$ Mbps. From simulation results, the ERICA+ faces the severe problem regarding the divergence of sources ACR as the effect from receiving all sources BRM cells during different averaging interval and the algorithm is per average interval computing [6] leading to growing unbound of switch queue length. For E-FMMRA algorithm, although the allocated rates converge to the expected value but the queue level at the bottleneck switch cannot be conducted to the desired region (QT and DQT is set to 50 and 1000 cells, respectively). For FRACA algorithm, the allocated rates converge very fast to the expected value with controllable switch queue length.

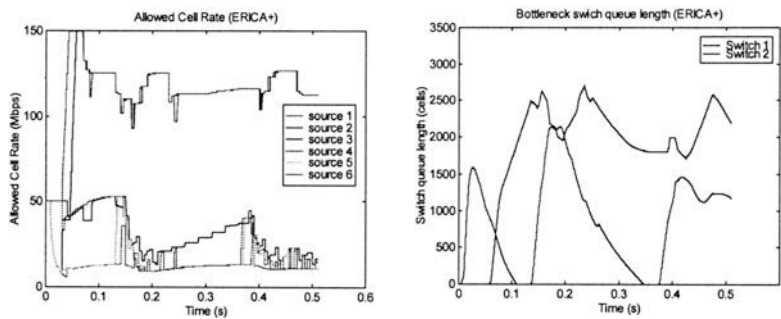
2. Parking-lot + link bottleneck + source bottleneck configuration



Parameters setting:

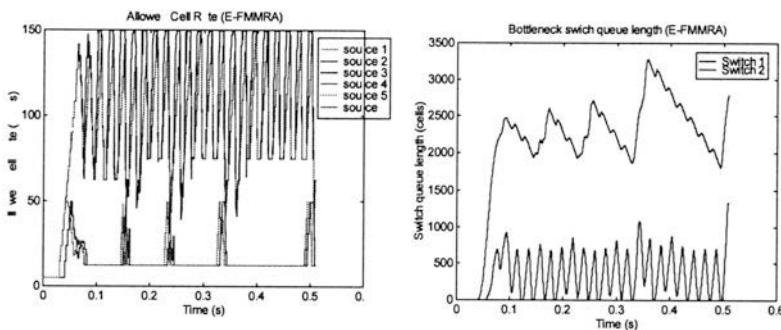
- Sources 1, 3 and 5 are persistent sources, PCR = 150.0 Mbps., MCR = 0.0 Mbps., ICR = 5.0 Mbps., Sources 2, 4 and 6 are bottlenecked sources (send data at rate 10 Mbps. all the time)
- All link propagation delay = 5 msec.

Figure 5. Parking-lot + link bottleneck + source bottleneck configuration



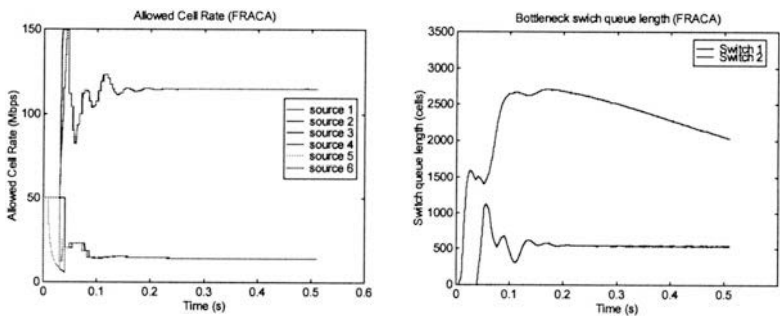
(a) Allowed Cell Rate in Mbps. (b) Bottlenecked switch queue in cells

Figure 6. Results for Parking-lot + link bottleneck + source bottleneck configuration (ERICA+)



(a) Allowed Cell Rate in Mbps. (b) Bottlenecked switch queue in cells

Figure 7. Results for Parking-lot + link bottleneck + source bottleneck configuration (E-FMMRA)



(a) Allowed Cell Rate in Mbps. (b) Bottlenecked switch queue in cells

Figure 8. Results for Parking-lot + link bottleneck + source bottleneck configuration (FRACA)

E-FMMRA faces the severe problems regarding divergence of sources ACR in source bottleneck configuration leading to oscillation of switch queue level in switch#1. ERICA+ performs worse with source bottleneck configuration. The reason is that ERICA+ algorithm uses CCR field in forward RM cell to compute the feedback ER. But for source bottleneck scenario the CCR field does not reflect the actual transmit rate. The following are the expected rates according to max-min fairness criterion.

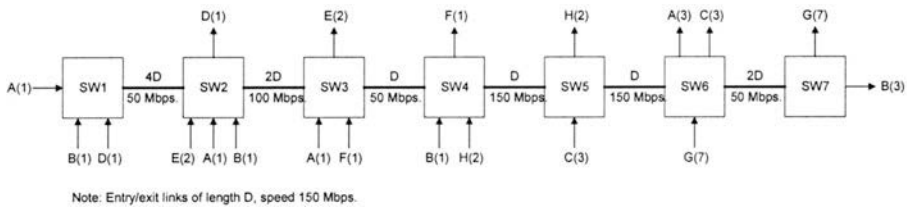
Source#1, #2, #5, #6 = 15.0 Mbps.

Source#3, #4 = 115.0 Mbps.

Recall that source#2, 4 and 6 send data at 10 Mbps. all the time but their ACR field (in FRM cell) are 15.0, 115.0 and 15.0 Mbps, respectively. In the scenario, the ACR field does not reflect the actual transmit rate of these sources.

From simulation results, ERICA+ computes wrong ER especially Source#1, #2, #5 and #6. Although source#3 and #4 ACR converge to the targeted value but the convergence time is very long. Moreover, the utilization of link 2 is not 100 percent all the time (correspond to queue level in switch 2). With the proposed algorithm, all sources ACR converge to the expected rate very fast. The queue level is actually conducted to the targeted region results in full link utilization all the time.

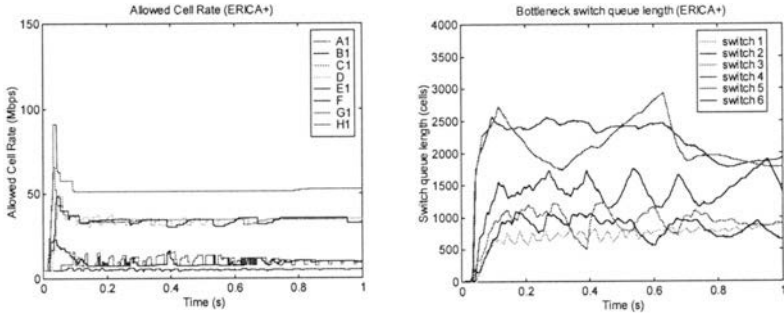
3. Generic Fairness Configuration – 2 (GFC-2)



Parameters setting:

- All sources are persistent: PCR = 150.0 Mbps., MCR = 0.0 Mbps., ICR = 5.0 Mbps.
- Link propagation delay (D) = 2.5 msec.

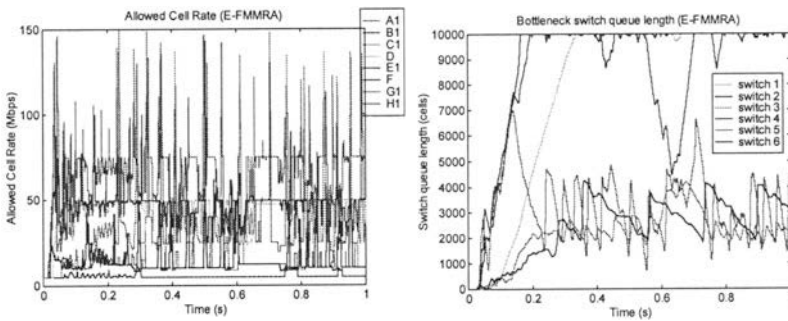
Figure 9. Generic Fairness Configuration –2 (GFC-2)



(a) Allowed Cell Rate in Mbps.

(b) Bottlenecked switch queue in cells

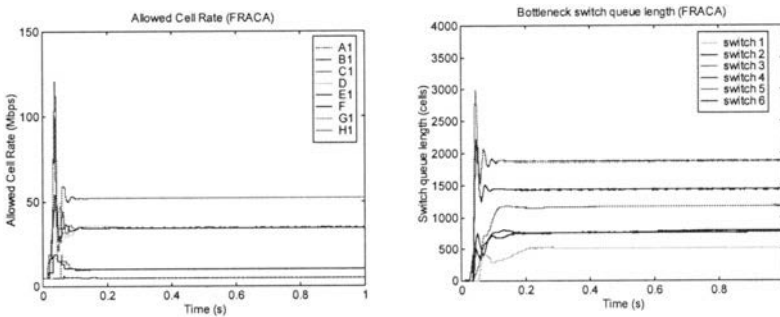
Figure 10. Results for GFC-2 configuration (ERICA+)



(a) Allowed Cell Rate in Mbps.

(b) Bottlenecked switch queue in cells

Figure 11. Results for GFC-2 configuration (E-FMMRA)



(a) Allowed Cell Rate in Mbps.

(b) Bottlenecked switch queue in cells

Figure 12. Results for GFC-2 configuration (FRACA)

GFC-2 is one of the contribution scenario in ATM Forum [5] for testing fairness and robustness of congestion avoidance algorithm. The following are the expected rate allocation according to max-min fairness criterion.

A VCs each get 1/4 of 40 Mbps.	= 10.0 Mbps.
B VCs each get 1/10 of 50 Mbps.	= 5.0 Mbps.
C VCs each get 1/3 of 105 Mbps.	= 35.0 Mbps.
D VC gets	= 35.0 Mbps.
E VCs each get 1/2 of 70 Mbps.	= 35.0 Mbps.
F VC gets	= 10.0 Mbps.
G VCs each get 1/10 of 50 Mbps.	= 5.0 Mbps.
H VCs each get 1/2 of 105 Mbps.	= 52.5 Mbps.

E-FMMRA algorithm diverges from the expected working region as the effect of multi-link rate and parking-lot configuration. The convergence time of ERICA+ is very long although the algorithm converges to the expected working region. Moreover, the queue level is cannot actually conducted to the desired value (5 msec. \approx 1766 cells in 149.76 Mbps. link) but it oscillates around the desired value instead as the effects from overload condition of the algorithm. For the proposed algorithm, sources ACR converges to the expected rate very fast with constant controllable queue level.

5. CONCLUSION

This paper presents a novel efficient and robust explicit rate congestion control algorithm for ABR service in ATM networks. The main goals for designing the proposed algorithm are convergence of sources ACR to the targeted working point according to max-min fairness with very fast transient response, 100% link utilization and controllable switch queue length with the simplest queue control function regardless of network condition and also should not affected by slow start phase of TCP application. We used 3 network configurations to illustrate performance of the proposed algorithm and also compared with ERICA+ and E-FMMRA, the well-known exact fair rate calculation algorithm. From comparison, the proposed algorithm works better than ERICA+ and E-FMMRA algorithm for all cases. Hence, our proposed algorithm should be appropriate exact fair rate calculation algorithm to implement in ATM switches working in real situation which network conditions vary simultaneously all the time.

6. ACKNOWLEDGEMENT

The authors are very thankful to Dr.Pichani Bodharamik, Dr.Somchai Jitapunkul and Dr.Narong Yoothanom for their encouragement and providing facilities for this research. Also heartfelt thanks are given to

Telecom Consortium Scholarship for giving part of financial support for research to the first author.

REFERENCE

1. "The ATM Forum Traffic Management Specification version 4.0", April 1996.
2. Arulambalam, and N. Ansari, "An Intelligent Explicit Rate Control Algorithm for ABR Service in ATM networks", IEEE International Conference on Communications, 1997.
3. Chiussi, Arulambalam, Ye Xia, Xiaoqiang Chen, "Explicit Rate ABR Scheme Using Traffic Load as Congestion Indicator", Proceeding of Sixth International Conference on Computer Communications and Networks, 1997.
4. Shiv Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal and Bobby Vandalore, "The ERICA Switch Algorithm for ABR traffic Management in ATM Networks", Submitted to IEEE/ACM Transaction on Networking, November 1997.
5. Robert J. Simcoe, "Test configurations for fairness and other tests", ATM Forum/94-0557, July 1994.
6. Tanun Jaruvitayakovit, Naris Rangsinoppamas and Prasit Prapinmongkolkarn "Improvement of ERICA+ Scheme using Adaptive Average Interval Algorithm", Applied Telecommunication Symposium'99, April 1999, San Diego, USA.
7. Arulambalam, X. Chen, and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM networks", IEEE communication magazine, 92-100, November 1996.
8. Bobby Vandalore, Raj Jain, Rohit Goyal, Sonia Fahmy, "Design and Analysis of Queue Control Function for Switch Scheme", ATM Forum/97-1087, December 1997.
9. Raj Jain, "Congestion control and traffic management in ATM networks: Recent advances and a survey", Computer Networks and ISDN Systems, 1995.
10. R. Onurval, "Asynchronous Transfer Mode Networks: Performance Issues", Artech House, 1994.