วิธีป้องกันความคับคั่งแบบระบุอัตราการส่งสำหรับบริการเอบีอาร์ในโครงข่ายเอทีเอ็ม

นายธนัญ จารุวิทยโกวิท

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2544

ISBN 974-03-1283-7

AN EXPLICIT RATE CONGESTION AVOIDANCE ALGORITHM FOR ABR SERVICE IN ATM NETWORKS

Mr. Tanun Jaruvitayakovit

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2001

ISBN 974-03-1283-7

Thesis Title	An Explicit Rate Congestion Avoidance Algorithm for ABR Service in ATM Networks
By	Mr. Tanun Jaruvitayakovit
Department	Electrical Engineering
Thesis Advisor	Professor Prasit Prapinmongkolkarn, D.Eng.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctor's Degree

…………………………………….. Dean of Faculty of Engineering
(Professor Somsak Panyakeow, D.Eng.)

THESIS COMMITTEE

……………………………………………….. Chairman
(Associate Professor Somchai Jitapunkul, Dr.Ing.)

……………………………………………….. Thesis Advisor
(Professor Prasit Prapinmongkolkarn, D.Eng.)

……………………………………………….. Member
(Supot Tiarawut, D.Eng.)

……………………………………………….. Member
(Assistant Professor Teerapat Sanguankotchakorn, D.Eng.)

……………………………………………….. Member
(Assistant Professor Lunchakorn Wuttisittikulkij, Ph.D.)

ธนัญ จารุวิทยโกวิท, นาย: วิธีป้องกันความคับคั่งแบบระบุอัตราการส่งสำหรับบริการเอบีอาร์ในโครงข่ายเอทีเอ็ม (An Explicit Rate Congestion Avoidance Algorithm for ABR Service in ATM Networks) อ.ที่ปรึกษา: ศ.ดร. ประสิทธิ์ ประพิณมงคลการ, 120 หน้า, ISBN 974-03-1283-7.

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการป้องกันความคับคั่งแบบระบุอัตราการส่งสำหรับบริการเอบีอาร์ในโครงข่ายเอทีเอ็ม บริการเอบีอาร์ในโครงข่ายเอทีเอ็มถูกออกแบบมาเพื่อรองรับการรับ-ส่งข้อมูลคอมพิวเตอร์ซึ่งมีคุณลักษณะที่คงทนต่อความล่าช้าในการรับ-ส่งได้ แต่จะไม่ทนต่อความสูญหายของข้อมูล ดังนั้นการรับ-ส่งข้อมูลแบบวงรอบปิดซึ่งมีการปรับอัตราการส่งของแหล่งกำเนิดตามสภาพความคับคั่งในโครงข่ายจึงสามารถนำมาประยุกต์ใช้กับบริการเอบีอาร์ได้เป็นอย่างดี นอกจากนั้นวิธีการป้องกันความคับคั่งที่เสนอจะต้องทำให้เกิดความเท่าเทียมกันในการรับ-ส่งข้อมูลของแหล่งกำเนิดต่าง ๆ ในระบบซึ่งมีลักษณะการเชื่อมต่อแบบต่าง ๆ กันในทางปฏิบัติได้ด้วย

ถึงแม้ว่าในปัจจุบันจะมีผู้เสนอวิธีป้องกันความคับคั่งแบบระบุอัตราการส่งสำหรับบริการเอบีอาร์อยู่บ้างแล้ว แต่ว่ายังมีข้อจำกัดที่สำคัญในทางปฏิบัติซึ่งทำให้วิธีป้องกันความคับคั่งแบบระบุอัตราการส่งที่มีอยู่มีประสิทธิภาพลดลงอย่างมากในการใช้งานจริง ตัวอย่างเช่น การทำงานในสภาพแวดล้อม (เช่นจำนวนการเชื่อมต่อมาก ๆ, รูปแบบของโครงข่าย, ลักษณะของข้อมูลขาเข้า และความล่าช้าทางเวลาของโครงข่าย) ที่มีการเปลี่ยนแปลงตลอดเวลา ตลอดจนการจัดสรรแบนด์วิธสำหรับการเชื่อมต่อที่ต้องการอัตราการส่งข้อมูลต่ำสุดที่ไม่เป็นศูนย์ จากการศึกษาพบว่าวิธีป้องกันความคับคั่งแบบระบุอัตราการส่งที่เป็นที่ยอมรับสองวิธีคือ ERICA+ (Explicit Rate Indication Congestion Avoidance) และ E-FMMRA (Enhanced Fast Max-Min Rate Allocation) จำเป็นที่จะต้องมีการปรับค่าตัวแปรต่าง ๆ ในการคำนวณเมื่อสภาพแวดล้อมต่าง ๆ ในโครงข่ายเปลี่ยนไป ดังนั้นจึงยังคงมีความจำเป็นที่จะพัฒนาวิธีป้องกันความคับคั่งแบบระบุอัตราการส่งที่มีประสิทธิภาพคงทนต่อสภาพแวดล้อมของโครงข่ายที่เปลี่ยนไปโดยไม่จำเป็นต้องมีการปรับค่าตัวแปรต่าง ๆ

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีป้องกันความคับคั่งแบบระบุอัตราการส่งที่มีชื่อว่า FRACA (Fast Rate Allocation Congestion Avoidance) เพื่อแก้ไขปัญหาต่าง ๆ ข้างต้น โดยวิธีป้องกันความคับคั่งที่เสนอมีความเข้ากันได้กับมาตรฐานของ ATM Forum นอกจากนั้นวิธีป้องกันความคับคั่งที่เสนอยังสามารถรองรับการคำนวณอัตราการส่งสำหรับการเชื่อมต่อที่ต้องการอัตราการส่งข้อมูลต่ำสุดที่ไม่เป็นศูนย์ได้ด้วย จากการจำลองการทำงานด้วยโปรแกรมคอมพิวเตอร์พบว่าวิธีป้องกันความคับคั่งที่เสนอมีประสิทธิภาพดีกว่า ERICA+ และ E-FMMRA ในทุกโครงข่ายที่ทดสอบ กล่าวคืออัตราการส่งที่คำนวณได้มีความถูกต้อง ส่งผลให้ระดับข้อมูลในหน่วยความจำอยู่ในระดับที่มีการออกแบบไว้ นอกจากนั้นยังพบว่าวิธีป้องกันความคับคั่งที่เสนอสามารถทำงานในสภาพแวดล้อมต่าง ๆ ที่เปลี่ยนไปได้เป็นอย่างดี ดังนั้นวิธีป้องกันความคับคั่งที่นำเสนอจึงเหมาะสมมากที่สุดที่จะใช้งานจริงในโครงข่ายเอทีเอ็ม จากการศึกษาเพิ่มเติมยังพบด้วยว่าวิธีป้องกันความคับคั่งที่เสนอสามารถเพิ่มประสิทธิภาพของการเชื่อมต่อจากจุดเดียวไปยังหลายจุดในบริการเอบีอาร์ด้วย ส่งผลให้วิธีการรวบรวมเซลล์ (Consolidation algorithm) แบบเรียบง่ายสามารถทำงานได้ดีเมื่อใช้งานร่วมกับวิธีป้องกันความคับคั่งที่นำเสนอ

วิทยานิพนธ์ฉบับนี้ยังทำการวิเคราะห์เชิงคณิตศาสตร์เพื่อแสดงให้เห็นว่าอัตราการส่งข้อมูลที่คำนวณโดยวิธีป้องกันความคับคั่งที่เสนอลู่เข้าหาความยุติธรรมแบบมากที่สุด-น้อยที่สุด (max-min fairness) ในกรณีที่แหล่งกำเนิดมีการส่งข้อมูลอย่างต่อเนื่อง (persistent source)

ภาควิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อนิสิต ....
สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่ออาจารย์ที่ปรึกษา ....
ปีการศึกษา	2544	

4071801121: MAJOR ELECTRICAL ENGINEERING

KEY WORD: ASYNCHRONOS TRANFER MODE / AVAILABLE BIT RATE/ RATE ALLOCATION ALGORITHM / FAST RATE ALLOCATION CONGESTION AVOIDANCE ALGORITHM

TANUN JARUVITAYAKOVIT, MR.: THESIS TITLE (AN EXPLICIT RATE CONGESTION AVOIDANCE ALGORITHM FOR ABR SERVICE IN ATM NETWORKS) THESIS ADVISOR: PROFESSOR PRASIT PRAPINMONGKOLKARN, D.ENG., 120 PP. ISBN 974-03-1283-7.

This dissertation is concerned with designing a comprehensive strategy for supporting the ABR traffic class in ATM networks. The ABR service class has been defined for reliable data service support in high-speed networks. Since data traffic is relatively delay tolerant, the use of feedback control schemes is appropriate. Moreover, it is necessary that the proposed algorithm provides fair resource distribution among competing connections and exhibits good scalability to many different network scenarios.

Although a number of ABR congestion control proposals have appeared, some important and critical issues still require further investigation. Some of these include operation in dynamic environments, containment of transient congestion effects and provision of MCR guarantees. From our investigation, performance of the well-known explicit rate congestion avoidance algorithms - the ERICA+ (Explicit Rate Indication Congestion Avoidance) and E-FMMRA (Enhanced Fast Max Min Rate Allocation) - depend significantly upon algorithm parameters selected which are affected by network conditions. The network conditions include a large number of traversing ABR sessions, network scenarios, traffic characteristics and system propagation delay. This limits both algorithms from working in the real environments that the network conditions change continually. Consequently, there is still a need for designing improved algorithm that addresses the above issues effectively.

The proposed FRACA (Fast Rate Allocation Congestion Avoidance) algorithm is designed to address limitations of current rate allocation algorithms. The proposed algorithm complies with the ATM Forum guidelines and implements the MCR plus equal share bandwidth fairness criterion. Simulation results indicate that the algorithm performs well in many scenarios. Scalability is good and queue sizes are also properly controlled. The performance also tolerates to continually change of the network conditions. Hence, the proposed algorithm should be appropriate to be implemented in the ATM switches working under real conditions. In addition, our proposed FRACA algorithm significantly improves the performance of point-to-multipoint ABR connections. As the result, the performance of a simple consolidation algorithm over the proposed FRACA algorithm is acceptable.

The convergence behavior of the FRACA algorithm is also analyzed mathematically. The analytical results indicate that the algorithm converges to the max-min fairness for the case that all sources are persistent.

Department	Electrical Engineering	Student’s signature ……………………..
Field of study	Electrical Engineering	Advisor’s signature …………………….
Academic year	2001	

Acknowledgements

I would like to thank my advisor, Professor Dr.Prasit Praminpongkolkarn for his continual guidance and support throughout my study. Without his support I could not bring this work to a completion. I have learned a lot from his insight into problem solving.

I am very thankful to Associate Professor Dr.Somchai Jitapunkul for his support especially during the 2 year period of my paper revision. Also many thanks are due to Associate Professor Dr.Somchai Jitapunkul, Assistant Professor Dr.Teerapat Sanguankotchakorn, Dr.Supot Tiarawut and Assistant Professor Dr.Lunchakorn Wuttisittikulkij for their review and comments of this dissertation.

I would like to thank my colleagues for their insightful discussion and warm encouragement. In particular, I would like to thank Dr.Naris Rangsinoppamas, Mr.Boonchoung Tansuthepverawongse. They have helped me in many steps of my research.

To my personal friends, I will not forget the friendship we made and the good time we had. I reserve specific mention for my friends Mr.Dheerasak Anantakul, Mr.Supachet Phempoonwatanasuk. Thank you very much for their good will and time. In particular, I would like to thank Dr.Chaodit Aswakul for his useful discussions and insightful comments during the revision of my paper.

The financial support I received was invaluable to my studies. I would like to acknowledge the National Science and Technology Development Agency (NSTDA), the Thailand Research Fund (TRF), the Graduated School of Chulalongkorn University and the fund of Prof. Dr.Prasit Prapinmongkolkarn. Thank you very much for making my student life much easier.

To my family, I could have never achieved this goal without the steadfast support from my parents, brothers and sisters. My parents have been unbelievably patient and supportive to me whatever the outcome.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

ABBREVIATION LIST

ABR - Available Bit Rate

ACR - Allowed Cell Rate

AI - Averaging Interval

ATM - Asynchronous Transfer Mode

BECN - Backward Explicit Congestion Notification

BRM - Backward RM cell

CAC - Connection Admission Control

CBR - Constant Bit Rate

CCR - Current Cell Rate

CI bit - Congestion Indication bit

CLP - Cell Loss Priority

DIR bit - Direction bit (RM cell)

EFCI - Explicit Forward Congestion Indicator

ER - Explicit Rate

ERICA+ - Explicit Rate Indication for Congestion Avoidance Schemes

GWF ERICA+ - General Weighted Fairness ERICA+

FIFO - First In First Out

E-FMMRA – Enhance Fast Max-Min Rate Allocation scheme

FRACA – Fast Rate Allocation Congestion Avoidance scheme

FRM - Forward RM cell

ICR - Initial Cell Rate

IETF – Internet Engineering Task Force

LAN - Local Area Network

MCR - Minimum Cell Rate

MSS - Maximum Segment Size

NI bit - No Increase bit

Nrm - Number of cells between FRM cells

PCR - Peak Cell Rate

PTI - Payload Type Indicator

QoS - Quality of Service

RDF - Rate Decrease Factor

ABBREVIATION LIST (Continued)

RIF - Rate Increase Factor

RM cells - Resource Management cells

RTT - Round Trip Time

TCP/IP - Transmission Control Protocol/Internet Protocol

TM4.0 - ATM Traffic Management Specification, version 4.0

VBR - Variable Bit Rate

VC - Virtual Circuit

WAN - Wide Area Network

CHAPTER 1
INTRODUCTION

1.1 Asynchronous Transfer Mode (ATM) Networks

Due to a variety of user needs, the trend of telecommunications in the globalization and information age is to transmit and receive very high speed integrated voice, data and image (e.g. IP telephony, video conference etc.) with a high quality and reliability. Towards this end, there are many broadband solutions available. Nonetheless, Asynchronous Transfer Mode (ATM) has been selected to deliver the Broadband Integrated Service Digital Network (B-ISDN) carrier service due to its capability to handle multiple levels of Quality of Services (QoS). Meanwhile, current and future networking technologies are expected to fulfil the goal of delivering QoS guarantees across integrated digital service networks. Each kind of traffic, i.e. voice data and video applications, has specific QoS requirements. QoS requirements are typically specified in terms of delivery of packets at the negotiated rate with minimal packet loss, delay and latency across the network. ATM networks have advantage of capability of multiplexing various data speed for bursty traffic as well as maintaining QoS guarantees.

ATM uses a point-to-point network architecture and transports data over Virtual Channels (VCs) using fixed size 53 bytes packets, called cells. As the result, the transmission time per cell is fixed and small causing low mean delay and low delay variance characteristics which are ideal for delay sensitive traffic such as packet-based voice and video transmission. In addition, the use of fixed cell size reduces the overhead of processing ATM cells and reduces the number of overhead bits required with each cell, thus enabling ATM to operate at high data rates. In addition, while setting up a connection on ATM networks, users can specify the following parameters related to the input traffic characteristics and the desired quality of service.

1. **Peak Cell Rate (PCR)** is the maximum instantaneous rate at which the user will transmit.

2. **Sustained Cell Rate (SCR)** is the average rate as measured over a long interval of time.
1. **Cell Loss Ratio (CLR)** is the percentage of cells that are lost in the network due to error and congestion and are not delivered to the destination, as defined in the following

$$\text{Cell Loss Ratio} = \frac{\text{Lost cells}}{\text{Transmitted Cells}}$$

Each ATM cell has a Cell Loss Priority (CLP) bit in the header. During congestion, the network first drops cells that have CLP bit set. Since the loss of CLP=0 cell is more harmful to the operation of the application, CLR can be specified separately for cells with CLP=1 and for those with CLP=0.

2. **Cell Transfer Delay (CTD)** is the delay experienced by a cell between network entry and exit point is called the cell transfer delay. It includes propagation delays, queuing delays at various intermediate switches, and service times at queuing points.
3. **Cell Delay Variation (CDV)** is a measure of variance of CTD. High variation implies larger buffering for delay sensitive traffic such as voice and video. There are multiple ways to measure CDV. One measure called "peak-to-peak" CDV consists of computing the difference between the (1-α)percentile and the minimum of the cell transfer delay for some small value of α.
4. **Cell Delay Variation Tolerance (CDVT)** and **Burst Tolerance (BT)**: For sources transmitting at any given rate, a slight variation in the inter-cell time is allowed. For example, a source with a PCR of 10,000 cells per second should nominally transmit cells every 100 μs. A leaky bucket type algorithm called "Generalized Cell Rate Algorithm (GCRA)" is used to determine if the variation in the inter-cell times is acceptable. This algorithm has two parameters. The first parameter is the nominal inter-cell time (inverse of the rate) and the second parameter is the allowed variation in the inter-cell time. Thus, a GCRA (100 μs, 10 μs), will allow cells to arrive no more than 10 μs earlier than their nominal scheduled time. The second parameter of the GCRA

used to enforce PCR is called Cell Delay Variation Tolerance (CDVT) and that used to enforce SCR is called Burst Tolerance (BT).

5. **Maximum Burst Size (MBS)** is the maximum number of back-to-back cells that can be sent at the peak cell rate but without violating the sustained cell rate. It is related to the PCR, SCR, and BT as follows:

$$\text{Burst Tolerance} = (\text{MBS} - 1)\cdot\left(\frac{1}{\text{SCR}} - \frac{1}{\text{PCR}}\right)$$

   Since MBS is more intuitive than BT, signaling messages use MBS. This means that during connection setup, a source is required to specify MBS. BT can be easily calculated from MBS, SCR, and PCR. Note that PCR, SCR, CDVT, BT, and MBS are input traffic characteristics and are enforced by the network at the network entry. CLR, CTD, and CDV are qualities of service provided by the network and are measured at the network exit point.

6. **Minimum Cell Rate (MCR)** is the minimum rate desired by a user.

ATM distinguishes itself from legacy networking technologies by providing a framework for various types of services and by specifying end-to-end QoS guarantees for different service classes. The following service categories have been defined by the ATM Forum [1]:

1. Real time service

There are various types of service categories available for ATM. The following will describe each service categories in brief:

1.1 Constant bit rate (CBR)

This category is used for emulating circuit switching. The cell rate is constant. Cell loss ratio is specified for CLP=0 cells and may or may not be specified for CLP=1 cells. Examples of CBR applications are telephone, video conferencing and television.

1.2 Real time Variable bit rate (rt-VBR)

This category allows users to send at a variable rate. Statistical multiplexing is used and so there may be a small nonzero random loss. Depending upon whether or not the application is sensitive to cell delay variation. For real-time

VBR, maximum delay and peak-to-peak CDV are specified during CAC process. An example of real time VBR is interactive compressed video.

2. Non real time service

2.1 Non real time variable bit rate (nrt-VBR)

This category is similar to real time VBR except that only the mean delay is specified during connection setup. An example of non real time VBR is real time reservation system such as air-ticket booking.

2.2 Available bit rate (ABR)

This category is designed for normal data traffic such as file transfer and e-mail. Although, the standard does not require the cell transfer delay and cell loss ratio to be guaranteed or minimized, it is desirable for switches to minimize the delay and loss as much as possible. Depending upon the congestion state of the networks, the source is required to control its rate. The users are allowed to declare a minimum cell rate, which is guaranteed to the virtual circuit (VC) by the network. Most VCs will ask for an MCR of zero. Those with higher MCR may be denied connection if sufficient bandwidth is not available.

2.3 Unspecified bit rate (UBR)

This category is designed for those data applications that desire to use any left-over capacity and are not sensitive to cell loss or delay. Such connections are not rejected on the basis of bandwidth shortage (no connection admission control) and not policed for their usage behavior. During congestion, the cells are lost but the sources are not expected to reduce their cell rate. Instead, these applications may have their own higher-level cell loss recovery and retransmission mechanisms. Examples of applications that can use this service are e-mail, file transfer etc.

Generally, the CBR and VBR service categories are assigned higher priority by the network switches and get a share of the link bandwidth firstly. The leftover capacity is utilized by the ABR and UBR services, respectively. Hence, the higher and lower priority traffic have to be queued separately at the queuing point as shown in Figure 1.1.

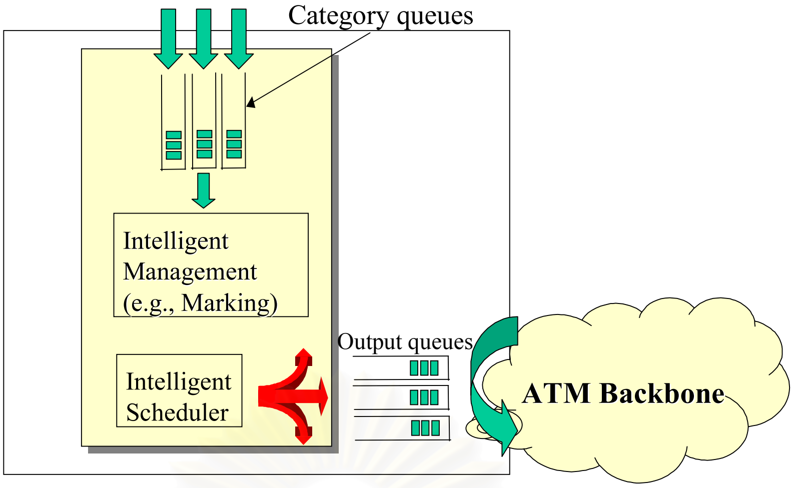


Figure 1.1: ATM switch modeling

Above queuing methodology simplifies the networks that queue the aggregate traffic in the per connection/session manner. For the per connection queuing, the queuing complexity increases linearly with the number of established connections which is difficult to handle in the real working environments.

This dissertation proposes a novel algorithm to solve the problem of supporting data applications by the ABR service category. Hence, the next section gives a briefly overview of the concept of the available bit rate (ABR) service category in ATM networks.

1.2 Congestion Management

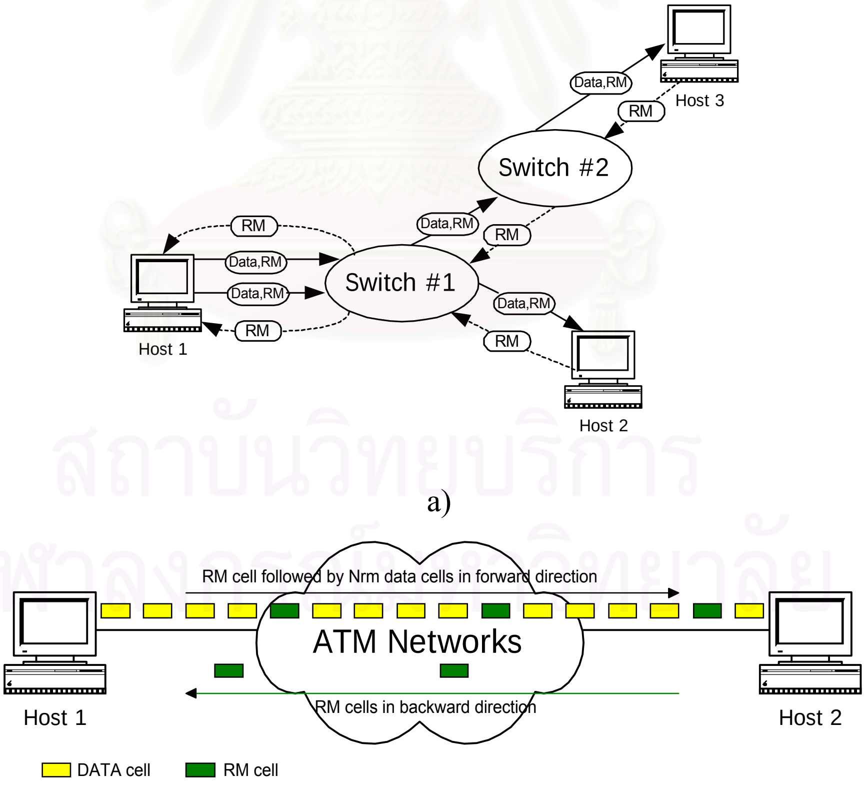
Generally, the network congestion happens whenever the total demand exceeds the available link capacity:

$$\sum_i Demand_i > Available\,Capacity$$

One method to avoid congestion in the network is to route according to the load level of links and to reject new connections if all paths are highly loaded. This technique is called Connection Admission Control (CAC). The “busy” tone on telephone networks is an example of CAC. CAC is effective only for medium duration congestion since once the connection is admitted the congestion may persist for throughout duration of the connection. For congestion time shorter than the duration of connection, an end-to-end control scheme can be used. For example, during connection setup, the sustained and peak rate may be negotiated. Later a leaky bucket algorithm may be used by the source or the network to ensure that the input meets the negotiated parameters. Such methods are open loop in the sense that the parameters cannot be changed dynamically if congestion is detected after negotiation. In a closed loop scheme, on the other hand, sources are informed dynamically about the congestion state of the network and are asked to increase or decrease their input rate in order to allurate the congestion.

1.3 Traffic Management for the ABR Service

Basically, the ABR service is designed for data applications. ABR service employs closed-loop for flow control. Initially, there were 2 models proposed for ABR closed-loop, credit-based and rate-based flow control. Figure 1.2 represents the model of the credit based and rate based flow control.



a)

b)

Figure 1.2: Models of

a) The credit based flow control and b) The rate based flow control

The credit based flow control [1, 12] is said to be a hop-by-hop flow control. The model consists of a per-link, per-VC window flow control. Each link consists of a sender node and a receiver node. Each node maintains a separate queue of each VC. The receiver monitors queue length of each VC and determines the number of cells that the sender can transmit on that VC. This number is called “credit”. Basically, the credit must be large enough to allow the whole link to be full at all times. In other words,

$$\text{Credit} \geq \text{Link cell rate} \times \text{Link round trip propagation delay}$$

After receiving a credit cell, the upstream switch (sender) sends an amount of the data cells in accordance with the credit field in the received credit cell. Hence, this method can avoid the networks from the cell loss problem but it cannot guarantee end-to-end QoS for each connection. Worse still, the computation complexity linearly increases with the number of established connections as a result of the computation of each per-VC connection. Hence, it cannot function efficiently in the real working environments.

For rate based ABR flow control, an ABR source sends control cells periodically. The corresponding destination sends back the control cells that are used by the intermediate switches to give the feedback information to the source. In the framework, an ABR source is allowed to send data up to the allocated rate by the network. Hence, the negotiated end-to-end QoS can be guaranteed across the network.

An extensive study was conducted by the ATM Forum’s Traffic Management committee to evaluate flow control strategies for the ABR service. Finally, a rate-based approach was overwhelmingly approved. The major reason for this was the large implementation complexities of credit-based schemes, which limited their scalability. Since the most current network transport technology is based on first-in-first-out (FIFO) queuing, the added per-connection requirements posed a significant obstacle to a gradual migratory approach to ABR services support.

1.4 Objectives

The major goal of the research is to design and analyze the performance of a novel explicit rate congestion avoidance algorithm for ABR service in ATM networks. The approaches to be considered are strictly at the cell-level. The proposed algorithm should fit into the specified explicit rate guidelines set by the ATM Forum [1], and must have significant advantages over existing rate-based proposals in the ATM Forum i.e. related to performance issues such as queue control, bandwidth fairness, scalability etc. Furthermore, the proposed algorithm should have an acceptable implementation complexity and be able to function well in a wide range of network conditions.

1.5 Scope of work

From our studies, the performance of the existing explicit rate congestion avoidance algorithms depend on the algorithm parameter settings which are affected by network conditions such as number of traversing ABR sessions, traffic characteristics and system propagation delay [11]. In other words, in order to maintain the algorithm performance the parameter settings have to be modified when network conditions change which cause the algorithms cannot function efficiently in the real working environments. Hence, the main goal for designing the novel algorithm is the tolerance of the changes in network conditions. In other words, the proposed algorithm should function well, i.e. achieve all designed goals, in all tested scenarios which certainly exist in the real working environments.

1.6 Thesis Organization

In this dissertation, related works in the area of ABR switch congestion avoidance algorithm are surveyed in Chapter 2. The drawbacks and limitations of each algorithm that cause the algorithms cannot function efficiently in the real working environments are stated. Chapter 3 proposes a novel rate allocation algorithm for point-to-point connection with zero Minimum Cell Rate (MCR) guarantees. The simulation results are employed to illustrate the performance of the proposed rate allocation algorithm compared with the existing algorithms, ERICA+ and E-FMMRA. In addition, the analytical proof is given to guarantee the convergence of the proposed algorithm.

Chapter 4 extends the proposed rate allocation algorithm to support non-zero MCR guarantee in ABR service. Chapter 5 evaluates the effects of point-to-point rate allocation algorithm on the performance of point-to-multipoint ABR connections. Chapter 6 concludes the dissertation.

CHAPTER 2
SURVEY OF ABR SWITCH CONGESTION AVOIDANCE ALGORITHMS AND THEIR RELATED PROBLEMS

In this chapter, the existing ABR switch congestion avoidance algorithm proposals are being evaluated. For each algorithm, the key technique is given. In addition, we will identify the key contributions and its drawbacks of each algorithm. This will lay a foundation for comparison with the Fast Rate Allocation Congestion Avoidance (FRACA) algorithm to be proposed and developed in this dissertation.

2.1 The Explicit Forward Congestion Indication (EFCI) Marking Scheme

The EFCI marking switch algorithm is the initial stage of the traffic management for ABR service in ATM networks. EFCI essentially uses a single bit feedback. The initial binary feedback algorithm used a “negative polarity of feedback” in the sense that RM cells are sent only to decrease the source rate (no RM cells are required to increase the rate). On the other hand, a “positive polarity of feedback” would require sending RM cells for increase but not for decrease. The problem with negative polarity is that if the RM cells are lost due to heavy congestion in the networks, the sources will keep increasing their load on the forward path and eventually overload the networks. The problem was fixed by using positive polarity. The sources keep decreasing their rate until they receive a positive feedback. However, the positive polarity was found to have a fairness problem [12]. Given the same level of congestion at all switches, the VCs travelling more hops have a higher probability of facing congestion than those travelling smaller hops. Hence, long path VCs have fewer opportunities to increase and are beaten down more often than short path VCs. The problem was called “beat down problem” [12].

In the next version of the binary feedback, the sources could increase and decrease their load in addition to the bit information in the received RM cell. The methodology would be called “bipolar”.

2.1.1 Key techniques

The methodology defines the binary feedback framework. EFCI is a bit located in the header of a ATM cell. Initially, the EFCI bit is cleared (set at 0) at the source. The congested switches set the EFCI bit to 1 (the intermediate switches detect the congestion via their switch queue level). The corresponding destination maintains an EFCI state per VC and set the Congestion Indication (CI) bit in the backward RM cell if the VC's EFCI state is set. On the other hand, the CI bit is cleared if the VC's EFCI bit is not set. The model of EFCI marking switch algorithm is shown in Figure 2.1.

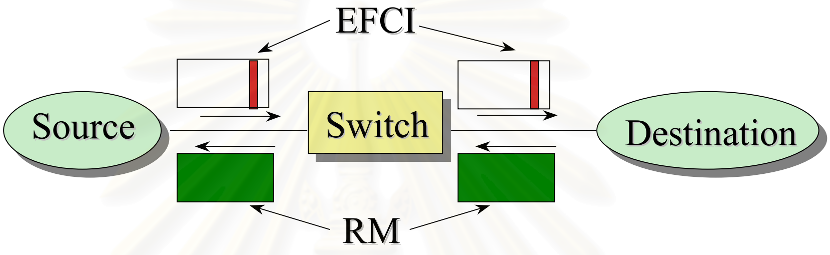


Figure 2.1: Model of the EFCI marking scheme

The source adjusts their rate in accordance with the information in the CI bit of the received backward RM cell. The methodology is bipolar in the sense that sources could increase their load if the CI bit in the received backward RM cells is clear. On the other hand, sources have to decrease their load if the CI bit is set.

In modern bit-based feedback framework, the "Intelligent Marking" technique [12] was proposed to improve the performance of the single-bit based feedback. The additional "No Increase" (NI) bit was included in the RM cell. Thus, there is more than one threshold of the switch queue length to indicate levels of congestion (corresponding to the additional NI bit). If the switch queue length is less than the low threshold (switch is not congested), the NI and CI bit in the RM cell are cleared. Hence, the source could increase its cell rate. On the other hand, if the switch queue length is between the low and high threshold (switch is lightly congested), the NI is set while the CI bit is cleared. The source remains its cell rate, respectively. Finally, if the switch queue length is more than the high threshold (switch is heavily congested), the NI and CI bit are set. Hence, the source has to decrease the cell rate. However, the "Intelligent Marking" technique could not able to address the beat down problem.

2.1.2 Characteristics of EFCI marking scheme

The contributions of the EFCI marking switch algorithm are as follows:

1. The bit-based feedback scheme firstly uses the queue length as a (single) metric for congestion detection. Using the queue length as a metric is a safe method to avoid congestion although it is not a principal parameter in the rate-controlled framework.

2. EFCI is idle for implementation because only one bit is used in the header. The feedback calculation is also typically simple.

The drawbacks of the EFCI marking switch algorithm are as follows:

1. The use of single bit notification cannot prevent the network from congestion. The reason is that all sources can send their data up to the negotiated PCRs if the CI and NI bits of the received BRM cells are clear. For the long propagation delay scenarios (i.e. Wide Area Network (WAN) or satellite ATM network), which the BRM cells take a long time to inform the congestion to all sources,

2. The system may take many round trips to converge to the steady state since a bit gives only two pieces of information (increase or decrease). Hence, the transient convergence period is long.

3. The steady state exhibits oscillatory behavior in terms of rate allocations and queue length. The reason for the behavior is that the control is based upon the queue length that is not a principal metric in the rate control networks.

4. EFCI is found to have a fairness problem. Given the same level of congestion at all switches, the VCs travelling more hops have a higher probability of having EFCI set than those travelling smaller number of hops. Consequently, the sources that locate near the corresponding destinations can send more data than the sources that locate far away. The problem is called “beat-down problem [12]”.

5. EFCI is sensitive to the parameter settings, for example Rate Increase Factor (RIF), Rate Decrease Factor (RDF) and level of QT and DQT at switches. The parameter settings for optimal response should be further studied.

6. The buffer requirement at switches is large and increases linearly with the number of traversing sessions.

2.2 The Explicit Rate (ER) Scheme

The explicit rate scheme is designed to solve the problems of the bit-based feedback schemes described in the previous section. The schemes attempt to allocate the fair rate, based on the available bandwidth and per-connection state information, to all sessions. Hence, the switch has to maintain a connection-based table to store per-connection information for computation process.

2.2.1 Key techniques

ATM Forum TM 4.0 [1] defined standard for source and destination behaviors. However, the switch behavior is left for switch manufacturer. When a source has received a permission, it begins to send user data cells at the Allowed Cell Rate (ACR). The ACR is initially set at Initial Cell Rate (ICR) and typically bounded between the negotiated Minimal Cell Rate (MCR) and the Peak Cell Rate (PCR). A ABR source will continue to send a control cell, named Resource Management (RM) cell, typically every sending $N_{rm} - 1$ data cells into the networks. The source places the ACR value in the Current Cell Rate (CCR) field of the RM cell and the rate it wishes to transmit cells (usually the PCR value) in the Explicit Rate (ER) field of the RM cell. The RM cells traverse across the network in the forward direction, named Forward RM (FRM) cells, to the destination. After receiving the FRM cells, the destination turns around and send the received FRM cells in backward direction, named Backward RM (BRM) cells. The intermediate switches compute the rate they can handle to prevent the network from congestion and convey the computed rate in the ER field of the received BRM cells. Upon receiving the BRM cells, the source adapts its ACR value follows the ER field carried in the BRM cells. Figure 2.2 shows model of the explicit rate scheme.

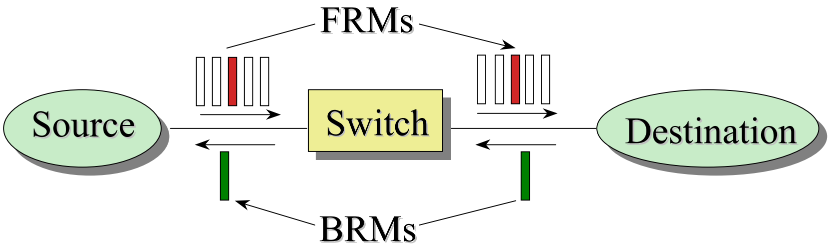


Figure 2.2: Model of the explicit rate scheme

In addition to providing a solution to the problems of the bit-based feedback framework, explicit rate schemes are attractive for other reasons. First, policy is straightforward. The switches can monitor the returning RM cells and use the rate directly in their policing algorithm. Second, the system converges to the optimal operating point quickly. Initial rate has less impact. Third, the schemes are robust against errors in RM cells as well as the loss of RM cells. The next RM cell carrying correct feedback will bring the system to the correct operating point in a single step.

2.2.2 Design Criteria

Generally, the design and implementation of a rate allocation algorithm should meet the following criteria:

- Fairness

Ideally, a congestion avoidance algorithm should equally divide the available bandwidth among sources which can utilize the bandwidth (in the scenario that all sources MCR are zero). A commonly used fairness criterion is max-min fairness, presented in [33]. The key idea of the max-min fair rate allocation is to allocate the fairly minimum rates to all sources causing the maximum utilization of the available bandwidth. [12] showed the simple procedure for finding the max-min rate allocations which can be formulated iteratively as follows:

1. Find the equal share for the connections on each link.
2. Find the connections with the minimum equal share.
3. Subtract this rate at the link and eliminate the connections with the minimum equal share.
4. Re-compute the equal share of each link in the reduced network.
5. Repeat procedures 2-4 until all connections are eliminated.

Recall that one of the main aims for ABR service is to fully utilize left-over bandwidth (from the higher priority service categories) and ABR is designed for handling data application whereby traffic characteristics are bursty in nature. Hence, the above procedures have to be modified in order to meet the goals for ABR service.

- Transient response

The transient response is almost as an important issue because the real world traffic is bursty. A fair rate allocation should take the minimum round-trip time to get close to the optimal rate. Hence, the allocated rates should rapidly converge from overload or underload states to the steady state.

- Steady state response

The steady state of a fair rate allocation is a state that the design goals have been achieved. The rate allocation should be able to converge to a steady state from any set of initial conditions with the minimum rate oscillation around the optimal rate.

- Robustness

A fair rate allocation algorithm should operate correctly in dynamic load changes even in the presence of sudden traffic changes and parameter mistunings. For the algorithms that rely on the tuning of parameter(s), any parameter that has not been set correctly may lead to significantly performance degradation.

- Interoperability

A fair rate allocation algorithm should be compatible with the source and destination rule defined in the standard body [1]. Hence, the algorithm can operate efficiently even though in multi-vendor environments. The algorithms that are not compatible with the standard or employ additional field(s) of the RM cell cannot operate anymore in the networks that other fair rate allocations exist.

- Implementation complexity

Generally, the rate allocation process based on a per-connection basis causes high implementation complexity. Hence, the required computations for a rate allocation algorithm should be kept to a minimum such that it can be implemented at a reasonable cost added to the switch design.

2.2.3 Existing Explicit Rate Congestion Avoidance Algorithm for ABR service in ATM networks

During a couple year, a number of explicit rate congestion avoidance algorithms for ABR service have been proposed. This dissertation focuses on the algorithms that are compatible with ATM Forum traffic management 4.0 standard. From our survey, the most efficient rate allocation algorithms are the Explicit Rate Indication Congestion Avoidance Plus (ERICA+) and the Enhance Fast Max-Min Rate Allocation (E-FMMRA) [4, 12].

2.2.3.1 The Explicit Rate Indication Congestion Avoidance Plus (ERICA+)

The ERICA and ERICA+ schemes were proposed by S. Kalyanaraman, R. Jain, R. Goyal, S. Fahmy and B. Vandalore [4] at Ohio-State University. The Explicit Rate Indication Congestion Avoidance plus (ERICA+) was further improved from the basic ERICA algorithm to achieve the max-min fairness [23]. The basic ERICA performs unfair behavior [21] if these following conditions are met

1) The load factor becomes one.
2) There are some connections which are bottlenecked upstream and
3) The source rate for all remaining connections is greater than the FairShare.

The detail techniques of the ERICA+ algorithm are shown in the next section.

2.2.3.1.1 Key techniques

The algorithm [4, 23] monitors the forward flow and gives the feedback in BRM cell. The feedback calculation may be performed when a BRM cell is received in the reverse direction. The algorithm calculates fair share rate that can be defined as

$$\text{FairShare} = \frac{\text{ABR Target Rate}}{\text{Number of active connections}} \qquad (2\text{-}1)$$

where ABR Target Rate is the multiplication result of the leftover bandwidth unused from the higher priority service classes and hyperbolic queue control function [f(Q)] which is defined as follows [4]:

$$f(Q) = \begin{cases} \dfrac{bQ_0}{(b-1)Q + Q_0} & \text{for } 0 \le Q \le Q_0 \\ \text{Max}\left(\text{QDLF}, \dfrac{aQ_0}{(a-1)Q + Q_0}\right) & \text{for } Q > Q_0 \end{cases} \qquad (2\text{-}2)$$

where a and b are fixed parameters set at be 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 at be 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 ERICA+ hyperbolic queue control function is shown in Figure 2.3.

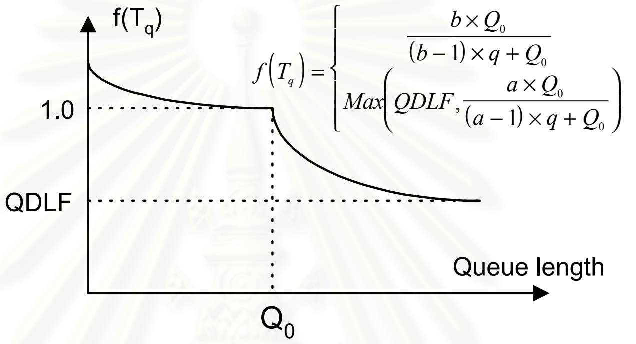


Figure 2.3: The ERICA+ hyperbolic queue control function

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 follows:

$$\text{VCShare} = \frac{\text{CCR}}{\text{Z}} \qquad (2\text{-}3)$$

where CCR (Current Cell Rate) is derived from the received forward RM cell (FRM) 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 sessions during current averaging interval and also MaxAllocPrevious which is the maximum value of every computed ER for all sessions during previous averaging interval. Upon reception of a BRM cell, the ER field is marked down as follows:

```
if ($Z > 1+\delta$)          ... where $\delta$ is a small value, typically set at 0.1
        ER in BRM cell = min (ER in BRM cell, max(fairShare, VCShare))
else    ER in BRM cell = min (ER in BRM cell,
                                  max(fairShare, VCShare, MaxAllocPrevious))
                                                                      (2-4)
```

2.2.3.1.2 Characteristics of ERICA+

The contributions of the ERICA+ switch algorithm are as follows:

1. The algorithm firstly investigates the use of the load factor to track down the actual transmitted traffic. Hence, the unconstrained sessions (the sessions that can utilize higher allocated rate) should be able to utilize the unused bandwidth by the constrained sessions (the sessions that bottlenecked elsewhere).

2. In the algorithm, the switch queue length is the additional parameter used in the rate allocation process. Consequently, the switch queue length is maintained in addition to the congestion avoidance.

The drawbacks of the ERICA+ switch algorithm are as follows:

1. The algorithm performance significantly depends on the parameter settings which is mainly affected by network parameters such as number of traversing ABR sessions, propagation delay, traffic characteristics and network scenario. For some scenarios, for example a large number of traversing ABR sessions or staggered TCP scenario, the allocated rates by the algorithm diverge from the max-min fair rate. Consequently, the switch queue level cannot be conducted to the designed region. Some researches (from ERICA+ authors) employed different values of parameter settings. For example, [4] recommends b (one of queue control parameters) to be 1 whereas [5] uses 1.05, [4] sets averaging interval to 5 msec but [21] recommends to be the minimum of time to receive 100 cells and 1 msec.

2. The use of CCR field in FRM cell in the rate allocation process cannot function efficiently in the scenario that the CCR filed does not reflect the actual transmit rate, i.e. source bottleneck and slow start phase of TCP applications. As the result, the algorithm performance is significantly degraded due to the error of rate allocation processes.

3. At steady state, the switch queue level periodically oscillates around the designed value in stead of conduction to the designed value as a result from the overload condition during the working region (the load factor is located between 1 and 1+δ). The switch queue behavior indicates that the allocated rates are not exactly the max-min fair rates.

4. In general, a queue control methodology is only an option for controlling switch queue level in the rate allocation algorithms. Hence, the performance of rate allocation process (i.e. achieving max-min fair rate) should not be affected by the queue control function. [9] investigated that the ERICA+ performance considerably depends on the genre of the queue control functions. Working along with step or linear queue control function, the ERICA could not converge to the max-min fair rate. In addition, our studies indicate that the allocated rates cannot converge without the designed hyperbolic queue control function.

2.2.3.2 The Enhance Fast Max-Min Rate Allocation (E-FMMRA)

The FMMRA and E-FMMRA were developed by Arulambalam, Chen and Ansari at New Jersey Institute of Technology and Bell Labs. [3, 12] showed that E-FMMRA outperform the basic ERICA. However, [3, 12] did not evaluate the performance of E-FMMRA compared to the ERICA+.

2.2.3.2.1 Key techniques

The algorithm computes the rate it can support which is known as advertised rate (γ). If a session 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 sessions. The ER field in the received RM cell is read and marked in both directions to speed up the rate allocation process [2]. The allocated rates to all non-bottleneck sessions are recorded as the maximum value of ER, denoted as ER_{max}. The algorithm computes ER_{adjust} every time the switch receives FRM and BRM cell.

$$ER_{adjust} = \frac{ER_{max}}{\text{Load factor}} \qquad (2\text{-}5)$$

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 the following relation when the switch receives BRM cell.

$$ER_{max} = (1 - \alpha)ER_{max} + \alpha \max(ER, ER_{adjust}) \tag{2-6}$$

where 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 at the maximum value between advertised rate and ER_{adjust}, that is

$$ER_{max} = \max(\gamma, ER_{adjust}) \tag{2-7}$$

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

$$\text{ER in FRM cell} = \min(\text{ER in FRM cell}, \max(\gamma, (1 - \beta)ER_{adjust})) \tag{2-8}$$

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

$$\text{ER in BRM cell} = \min(\text{ER in BRM cell}, \max(\gamma, (1 - \beta)ER_{adjust})) \tag{2-9}$$

$$\text{or} \quad \text{ER in BRM cell} = \min(\text{ER in BRM cell}, \max(\gamma, (1 - \beta)ER_{max})) \tag{2-10}$$

$$\text{or} \quad \text{ER in BRM cell} = \min(\text{ER in BRM cell}, \gamma) \tag{2-11}$$

The backward rate allocation (2-9 to 2-11) is employed upon the level of network congestion (switch queue length and load factor). If the switch is not congested, 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 (2-9). If switch is in moderate congestion condition and load factor is lower than or equal to unity, the switch computes ER according to (2-10). Finally if switch is in heavy congestion condition, the allocated rate is computed according to (2-11).

2.2.3.2.2 Characteristics of E-FMMRA

The contributions of the E-FMMRA switch algorithm are as follows:

1. Basically, the use of marking RM cells both forward and backward direction accelerates the convergence time.

The drawbacks of the E-FMMRA switch algorithm are as follows:

1. The calculation of feedback at the receipt of both the forward and backward RM cells increases the computation burden on the switch.

2. The source traffic characteristic mainly effects the performance of the algorithm. In the case of source-bottleneck scenario, source sends its traffic at the rate below the allocated rate, the algorithm diverges from the targeted working point as a consequence of setting too short averaging interval (as recommended in [2, 3]) and updating both FRM and BRM cell every time the switches receive. The reason for setting short averaging interval in [2, 3] is that the algorithm processes every received FRM and BRM cells. But setting short averaging interval, itself, cannot average aggregate input traffic rate correctly resulting in divergence of allocated rates.

3. The algorithm suffers from achieving convergence in parking lot or multi-link rate configuration as a result of marking both FRM and BRM cell, for example in the scenario that consists of both non-bottleneck and bottleneck sessions traversing a bottleneck switch. During startup condition, all sessions are classified into non-bottleneck grouping then the computed advertised rate is the fairshare among all ABR sessions. The advertised rate is assigned to all received FRM cells. After receiving BRM cells of the sessions which are bottlenecked elsewhere, i.e. the ER field in the received BRM cell is lower than the advertised rate, these sessions are marked as bottleneck sessions and their bottleneck bandwidth are then recorded. The algorithm computes a new advertised rate for non-bottleneck sessions. This new advertised rate is always greater than the old rate, then all sessions (including the expected non-bottleneck sessions) are marked as bottleneck sessions resulting in incorrect grouping between bottleneck and non-bottleneck sessions leading to wrongly allocated rates.

4. The algorithm cannot actually conduct the bottleneck switch queue level to the targeted region, i.e. between QT and DQT. For example, in the scenario that all sessions are non-bottleneck sessions – all sessions can utilize the advertised rate. In the scenario, advertised rate is the fairshare among all ABR sessions. If the sources have high Initial Cell Rate (ICR) compared to the max-min rates then the bottleneck switch queue level is greater than the high threshold. But the allocated rate to all

sessions is the advertised rate then the bottleneck switch queue length could not be conducted to the desired region.

5. Marking FRM cell sometimes causes under utilization in the network especially in the long propagation delay configuration i.e. satellite ATM networks. After marking FRM cell, if there is more bandwidth available (left over from higher priority category) in the network but the switches cannot allocate the higher ER in the BRM cell.

2.3 Summary

The former rate allocation algorithms were aimed to address the max-min fairness (beat down) problem in the EFCI switch while the later schemes addressed the speed of convergence and the implementation complexity issues. From our studies, however, the well-known rate allocation algorithms, ERICA+ and E-FMMRA, have some limitations that cause both algorithms cannot function efficiently in the real working environments. It is, therefore, proposed in this dissertation in subsequent Chapter 3 a Fast Rate Allocation Congestion Avoidance (FRACA) algorithm to solve the inherent limitations of both ERICA+ and E-FMMRA.

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

The FRACA algorithm is developed to overcome the limitations of the existing ABR congestion avoidance algorithms sited in the last chapter. The main design goals are the accurate average of aggregate traffic and fair rate allocation according to max-min fairness while exhibiting fast transient response in a wide range of network conditions, i.e., a large number of traversing ABR sessions, network scenarios, propagation delay and traffic characteristics. Since real networks are in a transient state most of the time, hence a rate allocation algorithm deployed in real-world switches need to perform well under both transient and steady state conditions. This chapter is organized as follows. Section 3.1 describes the concept of the FRACA algorithm. The simulation results and performance evaluation of common network scenarios are described and illustrated in Section 3.2. Section 3.3 investigates the performance of the FRACA algorithm in the more complex scenarios compared with the ERICA+ and E-FMMRA. Finally, the FRACA convergence analysis is given in Section 3.4.

3.1 The FRACA algorithm

The key steps of the FRACA algorithm are as follows. In order to concentrate on the bottleneck status of the given switch that a session traverses, a session is marked as a *non-bottlenecked session* if it cannot utilize the assigned rate allocation that is initially set at the fairshare. In other words, a non-bottlenecked session is a session that the Advertised Rate (AR) does not limit its traffic rate. As a result, it is bottlenecked elsewhere. On the other hand, a session is marked as a *bottlenecked session* if it can utilize higher rate allocation. Therefore, a bottlenecked session is a session that the AR or the maximum value of allocated rate (ER_{max}) limits its traffic rate. The pseudo-code for FRACA algorithm is as follows:

Parameters

C = Link capacity

$\Re$ = ABR capacity

$\Re'$ = Non-ABR (higher priority) capacity

$\Re_n$ = Total non-bottlenecked capacity

$\beta_{s(i)}$ = Bottlenecked state of VCI i

$\beta_{c(i)}$ = Non-bottlenecked capacity of VCI i

ER_{max} = Maximum value of allocated rates

N_b = Number of bottlenecked sessions

α = Averaging factor which is set at 1/8

RM→ER = ER field in RM cell

Adjusted ER = The value of ER for queue control option

When a BRM cell of VC i is received:

```
If ($\beta_{s(i)}$ = 1) then                                  /* bottlenecked session*/
        $RM \rightarrow ER = \min(RM \rightarrow ER, \max(AR, \frac{ER_{max}}{\rho}))$        (3-1)
Else                                                         /* non-bottlenecked session*/
        $RM \rightarrow ER = \min(RM \rightarrow ER, AR)$                                  (3-2)
$ER_{max} = (1.0 - \alpha) \times ER_{max} + \alpha \times \max(RM \rightarrow ER, \frac{ER_{max}}{\rho})$     (3-3)
If ($AR \leq RM \rightarrow ER$) then                         /* This is a bottlenecked session*/
        $N_b = N_b - \beta_{s(i)} + 1$                                               (3-4)
        $\beta_{s(i)} = 1$                                                           (3-5)
        $RM \rightarrow ER = RM \rightarrow ER$ + adjusted ER                         (3-6)
        $\Re_n = \Re_n - \beta_{c(i)}$                                               (3-7)
        $\beta_{c(i)} = 0$                                                           (3-8)
Else                                                         /* This is a non-bottlenecked session*/
        $N_b = N_b - \beta_{s(i)}$                                                   (3-9)
        $\beta_{s(i)} = 0$                                                           (3-10)
        $\Re_n = \Re_n - \beta_{c(i)} + RM \rightarrow ER$                           (3-11)
        $\beta_{c(i)} = RM \rightarrow ER$                                           (3-12)
```

If ($N_b > 0$) then

$$AR = \frac{\Re - \Re_n}{N_b} \tag{3-13}$$

Else $AR = AR + (\Re - \Re_n)$ (3-14)

When averaging interval is expired:

$$\Re = C - \Re' \tag{3-15}$$

$$\rho = \frac{\text{ABR Input Rate}}{\Re\ \text{Qfactor}} \tag{3-16}$$

$$\text{adjusted ER} = \frac{(\text{Qfactor} - 1)\Re}{N_b} \tag{3-17}$$

The mechanism of FRACA algorithm can be explained as follows. Upon receiving a BRM cell, the bottlenecked session will keep the minimum between the ER in RM cell and the maximum between the AR and the ratio of ER_{max} to ρ as the explicit rate according to (3-1). On the other hand, the non-bottlenecked session will keep the minimum between the ER in RM cell and the AR as the explicit rate according to (3-2). In the next step, the ER_{max} is exponentially averaged as by (3-3). After assigning the explicit rate, the session is re-grouped into either the bottlenecked or non-bottlenecked session. With regard to the definition, if the explicit rate is constrained by the AR, then the session is grouped into the bottlenecked session. Therefore, (3-4) and (3-5) updates the number of bottlenecked sessions and the bottlenecked state, respectively. Equations (3-7) and (3-8) show the update of the total non-bottlenecked capacity and the non-bottlenecked capacity of the session, accordingly. On the other hand, the session is grouped into the non-bottlenecked session if the explicit rate is not constrained by the AR. Then the number of bottlenecked sessions, the bottlenecked state, the total non-bottlenecked capacity and the non-bottlenecked capacity of the sessions are updated by (3-9)-(3-12), respectively. Finally, the AR is re-calculated from (3-13) and (3-14) by using the updated total non-bottlenecked capacity and the number of bottlenecked sessions.

After the averaging interval has expired, the working steps of the algorithm are as follows. Firstly, the ABR capacity is calculated by subtracting the non-ABR (higher priority) capacity from the link capacity as shown in (3-15). Secondly, the load factor is computed according to (3-16). Finally, the "adjusted ER" is calculated according to (3-17). The objective of using the "adjusted ER" is for queue control option. Remember that the rate of the bottlenecked session is constrained by the AR or ER_{max}. Consequently, the switch queue length is only affected by the allocated rates of the bottlenecked sessions. Using the allocated rates of the bottlenecked sessions as shown in (3-6) guides the queue length to be rapidly conducted to the desired level with oscillation-free allocated rates.

The FRACA algorithm runs exponential average of the ER_{max} to avoid the divergence of allocated rates when all sources receive BRM cells during different averaging interval in the per averaging interval rate allocation algorithm [7]. FRACA algorithm applies count-based averaging interval. The timer will be expired if the amount of received ABR cells reach a threshold, which is set at 1500 cells for an STM-1 link in our simulations. In addition, the counter number may be scaled by the ratio of STM-1 speed to a given bottleneck link speed for faster or slower links. However, a too low value of counter number causes the algorithm to inaccurately average the aggregate traffic. This results in the oscillation of allocated rates when the aggregate traffic is not consistent. Therefore, it is recommended to use a particular averaging interval at a link speed slower than DS3 (45 Mbps.), for example, the averaging interval of 200 cells for E1 link and 250 cells for E3 link. Setting an appropriate length of averaging interval causes the accurate averaging of aggregate traffic that results in the fast convergence according to the max-min fairness criterion. Moreover, the FRACA algorithm gives a single feedback per session in an averaging interval to stabilize the system.

For queue control, Qfactor is a fraction calculated from the step queue control function defined in Table 3.1. The queue control objective is to maintain switch queue length between a low level (set at 500 cells) and a high level (set at 2000 cells). If the switch queue length is lower than the low threshold, Qfactor is a little greater than unity. Hence, the allocated rates of the bottlenecked sessions are greater than the max-min rates in order to build the queue level up to the desired region. On the other hand, if the switch queue length is greater than the high threshold, the allocated rates of the bottlenecked sessions are lower than the max-min rates for draining the queue level. Finally, if the switch queue length is conducted to the desired level, the allocated rates of all sessions are the max-min fair rates resulting in well regulated buffer occupancy.

Table 3.1. Step queue control function

Switch queue length (cells)	Qfactor
0 < Q < 500	1.02
500 < Q < 2000	1.00
2000 < Q < 3000	0.98
3000 < Q < 4000	0.95
4000 < Q < 5000	0.9
5000 < Q < 6000	0.8
Q > 6000	0.6

The advantage of keeping non-zero queue level is maximizing the link utilization while working with bursty VBR traffics. It is because the queue will drain continuously when VBR traffics immediately change from on-phase to off-phase. As a result, there is no need to wait for BRM cells that will take time to inform the ABR sources.

3.2 Performance Evaluation of the FRACA algorithm

This section provides simple benchmarks to test the performance of the proposed FRACA algorithm. We present the set of experiments conducted to ensure that the FRACA meets all the requirements of switch algorithm. The results are presented in the form of four graphs for each configuration:

1. Graph of allowed cell rate (ACR) in Mbps over time for each source.
2. Graph of ABR queue lengths in cells over time at each switch.
3. Graph of link utilization (as a percentage) over time for each link.
4. Graph of number of cells received at the destination over time for each destination.

We will examine the efficiency, the fairness of the algorithm, its transient and steady state performance and finally its adaptation to variable capacity and various source traffic models.

3.2.1 Parameter setting

Unless otherwise noted, the following parameter values are hold for all simulations:

1. All links have a bandwidth of 149.76 Mbps (STM-1 accounted for SDH overhead).
2. All Local Area Network (LAN) links are 1 Km long and all Wide Area Network (WAN) links are 1000 Km long.
3. All sources, including VBR sources are deterministic, i.e., their start/stop times and their transmission rates are known.
4. All source Initial Cell Rates (ICR) are set at 10.0 Mbps (beside the scenario that indicates especially).
5. Each scenario simulation time is set at 500 msec., beside the scenario that indicates especially.

3.2.2 Efficiency and Queue control

To evaluate the efficiency and queue control methodology, we use a multiple source configuration. The simplest configuration is the two-source scenario, where two sources share a link as illustrated in Figure 3.1. At steady state, each source must converge to half of the link rate, which is the max-min optimal allocation, and the bottleneck switch (switch#1) queue length should be conducted to the designed region.

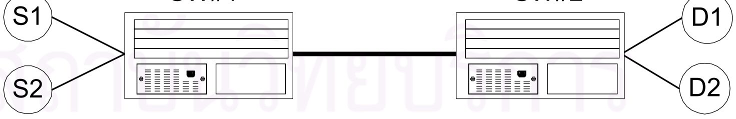


Figure 3.1: Two-source (LAN) scenario

3.2.3 Fairness

We use two scenarios to study the fairness of the FRACA algorithm: the parking lot scenario and the upstream scenario (see Figure 3.2 and 3.3, respectively). The parking-lot scenario is derived from car parking lots, which consist of several parking areas connected via a single exit path. Congestion occurs as cars exiting from each parking area try to join the main exit path. In other words, the link between switch#1 and 2 in Figure 3.2 is the bottlenecked link. Hence, the max-min fair rate to all traversing sessions is the fairshare of the bottlenecked link.

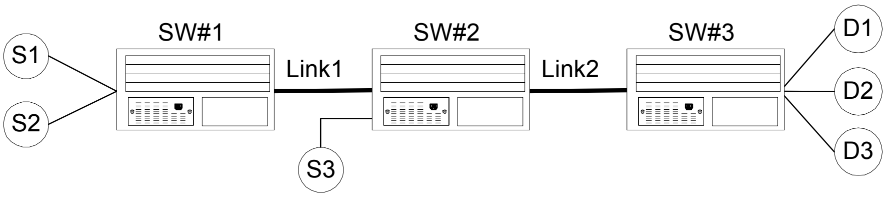


Figure 3.2: Parking-lot (LAN) scenario

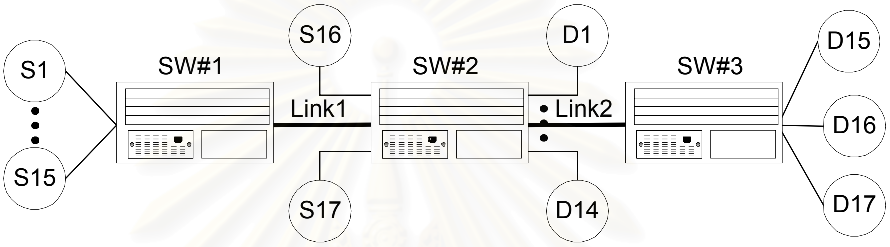


Figure 3.3: Upstream (WAN) scenario

In order to test the effects of sources ICR and propagation delay time to the performance of a rate allocation algorithm, the upstream scenario in Figure 3.3 is used. In the scenario, the ICRs of source 16 and 17 are different (set at 50.0 and 70.0 Mbps, respectively). Moreover, the scenario is evaluated in WAN environment. From the scenario, the second link is shared by 3 VCs, i.e. VC_{15}, VC_{16} and VC_{17}. Because there are 15 VCs on the first link, VC_{15} is bottlenecked to a throughput of 1/15 the link rate (approx. 10 Mbps). Consequently, VC_{16} and VC_{17} should converge to 7/15 of the second link rate (approx. 70 Mbps). The scenario is named an upstream scenario because the bottleneck link is the first link (upstream link).

3.2.4 ACR retention

ACR retention is the problem that occurs when sources are not able to fully use their rate allocations. For example, the input to the ATM end-system can be steady but have a rate lower than its allowed cell rate. In the situation, the switches reallocated the unused capacity to the other sources that are unconstrained. However, if the ACR retaining sources suddenly use their allocations, an overload situation occurs. Hence, the rate allocation algorithm should rapidly detects the overload and gives the appropriate feedback asking sources to decrease their rate. The ACR retention in Wide Area Network scenario is shown in Figure 3.4.

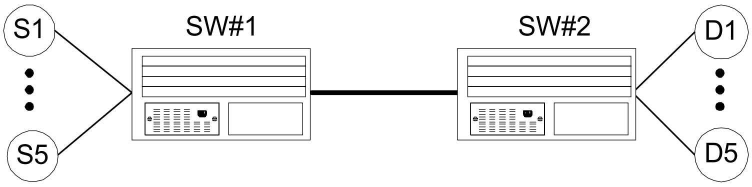


Figure 3.4: ACR retention (WAN) scenario

From Figure 3.4, there are 5 VCs sharing a bottleneck link. Initially, all sources cannot send send at a rate of more than 10.0 Mbps (irrespective of their ACRs). After 100 msec, all the sources suddenly start sending at their full allocations. Consequently, the bottleneck queue level should rapidly grows during the transient period. After overload detection, the bottleneck switch should reallocate the optimal rates to all sources in order to bringing the queue to the desired region and the allocated rates have to be the max-min rate as well.

3.2.5 Adaptation to variable ABR capacity

Constant Bit Rate (CBR) and Variable Bit Rate (VBR) service classes have a higher priority than the ABR service. In cases of VBR traffic, the ABR capacity becomes a variable quantity. The two ABR plus one VBR scenario in Wide Area Network is used to demonstrate the behavior of the FRACA in the presence of VBR traffic. A deterministic VBR source is used whose peak cell rate is 85.0 Mbps (57% of the link capacity). In addition, the on/off periods of 2 msec (active for alternating periods of 2 msec with 2 msec inactive periods in between) and 10 msec are used to evaluate the adaptation of the FRACA algorithm. The scenario is shown in Figure 3.5.

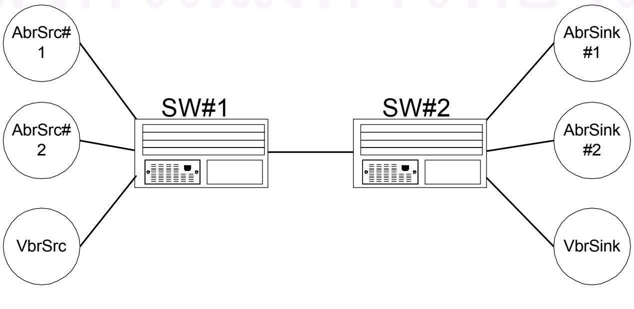


Figure 3.5: Two ABR plus one VBR (WAN) scenario

In the scenario, the FRACA algorithm should be able to detect the continually change in the available ABR capacity and gives the appropriate feedback to the sources. When the VBR source is active, the ABR sources should rapidly decrease their rate. On the other hand, the ABR sources should quickly increase their rate if the VBR source is inactive in order to keep the bottleneck link fully utilized.

3.2.6 Adaptation to various source traffic models

In all the previous experiments, the ABR sources are assumed to be persistent or uniform bottlenecked (that only match with the rate-limited phase of the TCP applications as mentioned in the last chapter). Then, it is essential to examine the performance of the FRACA with TCP applications that are actually used in the real working environments. TCP applications behaviour like bursty sources with non-uniform on-off period during their exponential phase (windows limit phase). The 20 TCP sources in Wide Area Network scenarios is employed to evaluate the performance of the FRACA in this issue. The scenario is shown in Figure 3.6.

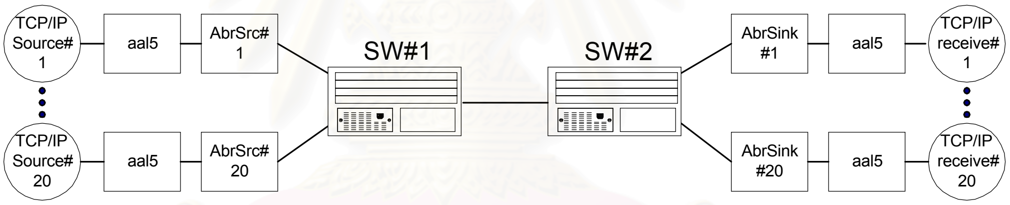


Figure 3.6: 20 TCP over ABR (WAN) scenario

3.2.7 Working with a low-speed line environment

The scenario is employed to evaluate the performance of the proposed FRACA algorithm under a low-speed line environment for an access network. The configuration is shown in Figure 3.7.

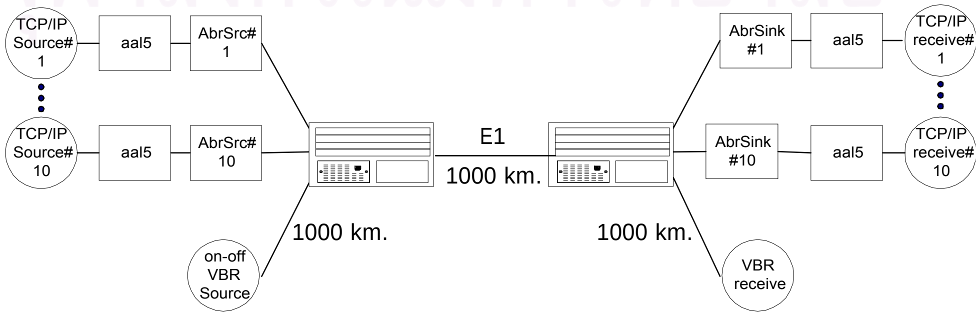
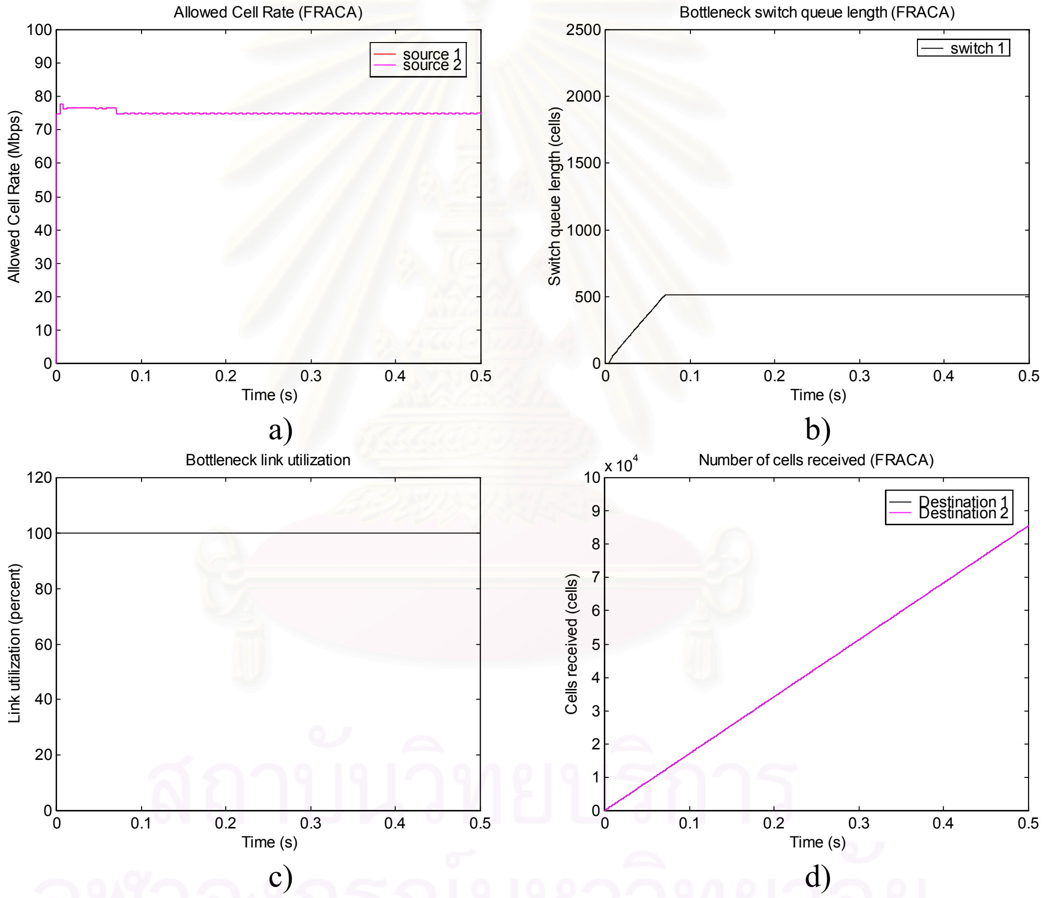


Figure 3.7: 10 TCP plus VBR over E1 link scenario

In the configuration, 10 TCP sources share an E1 link (2.048 Mbps). In addition, there is a bursty VBR traffic sharing the bottlenecked link. The VBR source is a Markov Modulated Bernoulli Processes (MMBP) source with mean duration of on/off time equal to 5 msec, aggregate VBR traffic is approximate 1.37 Mbps (2/3 of the link capacity) during on period. The simulation time is 5 seconds. For scalability, the algorithm should function efficiently though working with a low-speed line.

3.2.8 Performance results

This section gives the performance results of the FRACA algorithm under the scenarios mentioned above.

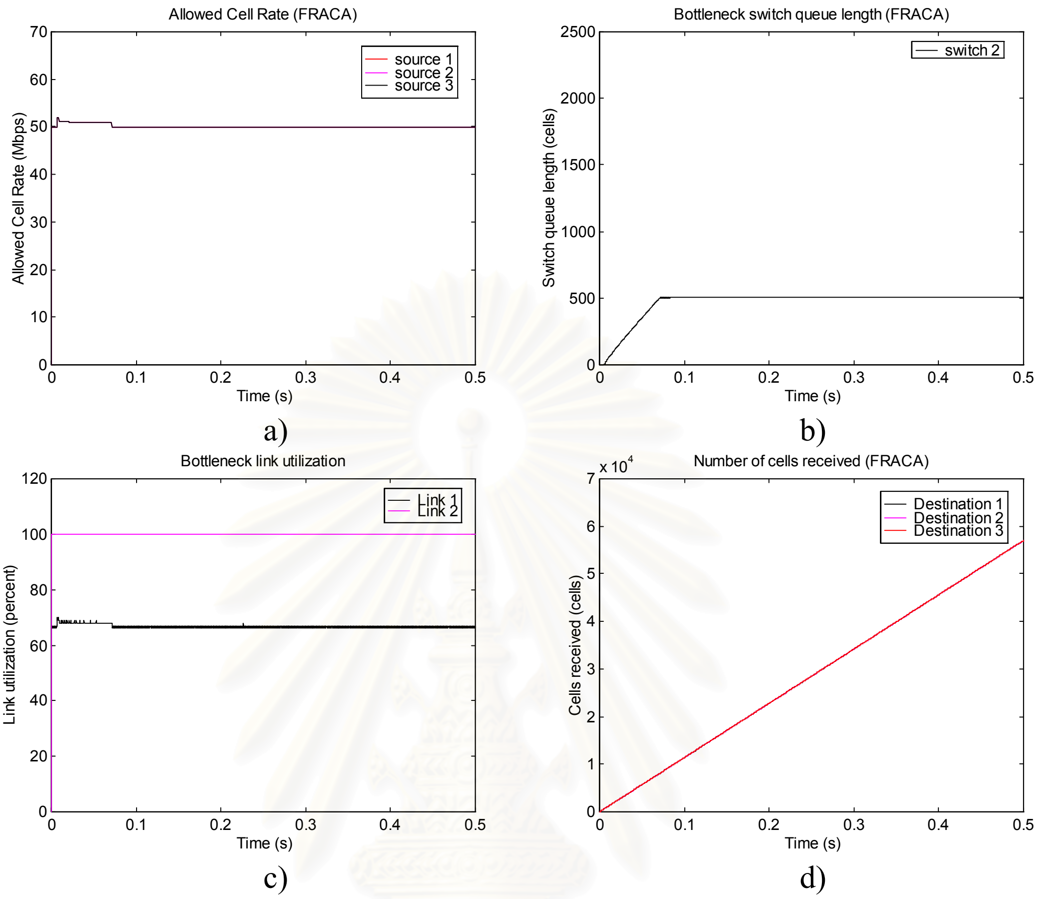


a) b) c) d)

Figure 3.8: Results for two source (LAN) scenario

Figure 3.8 indicates that the sources rapidly converge to their optimal rates and the switch queue length is conducted to the target level. The allocated rate during 0-80 msec is a little grater than the max-min rate. This is because the switch queue has to be built up to the designed level. However, the max-min fair rate is achieved after the switch queue reach the designed level. In addition, the number of cells received at each destination in Figure 3.8 d) illustrates that the fairness is achieved.

Figure 3.9 shows the results for parking-lot (LAN) scenario.



a)

b)

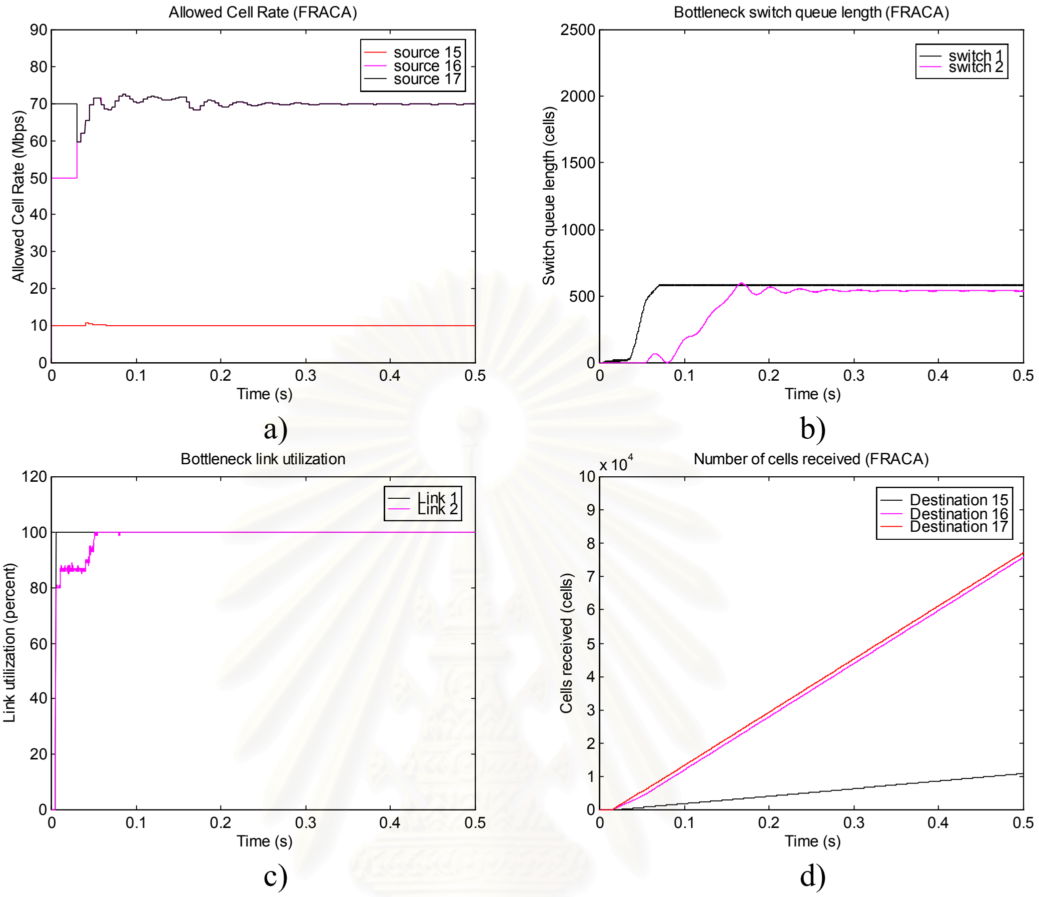
c)

d)

Figure 3.9: Results for Parking-lot (LAN) scenario

It is obvious from Figure 3.9 a) that the allowed cell rate of all sources rapidly converges to the max-min fair rates. The transient period is very short. The steady state behavior is good. The bottlenecked link is fully utilized while the switch queue is conducted to the target region (Figure 3.9 b) and c)). In addition, the fairness issue is ideally achieved according to Figure 3.9 d).

The results for upstream (WAN) scenario are shown in Figure 3.10.

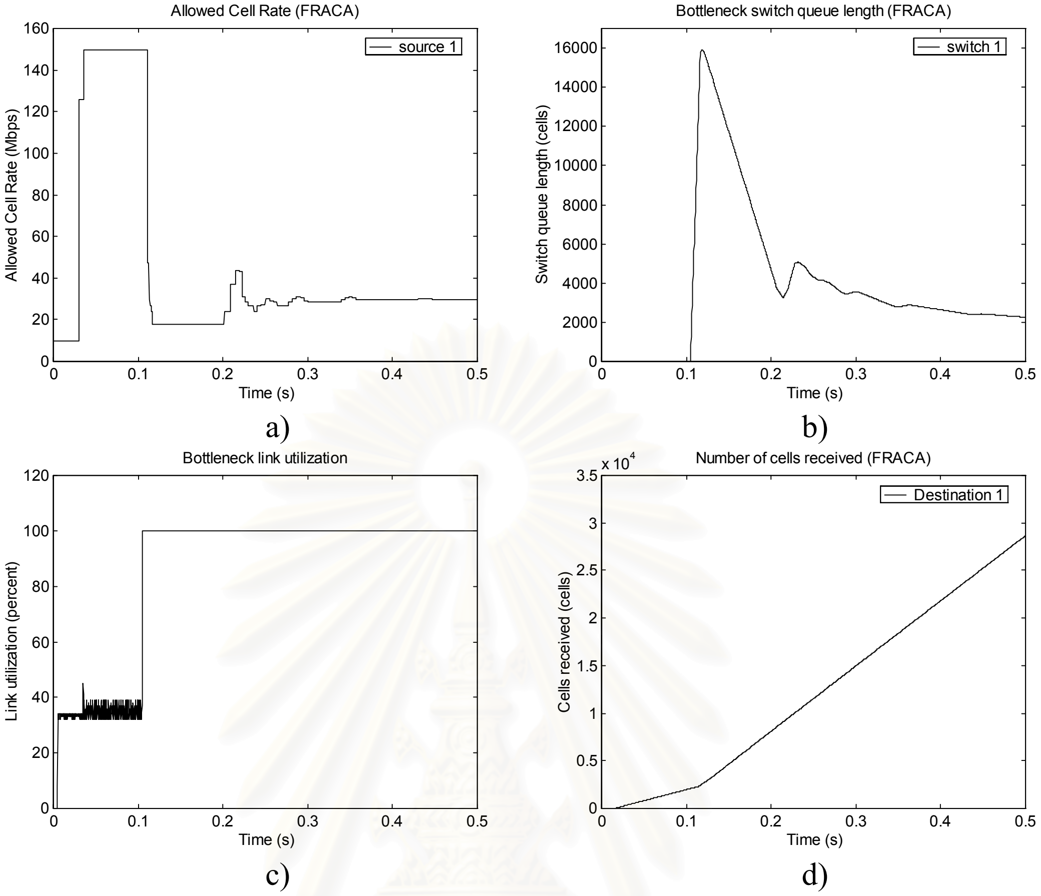


a) b) c) d)

Figure 3.10: Results for Upstream (WAN) scenario

Figure 3.10 a) indicates that the FRACA algorithm converges to the max-min rate allocation regardless of the initial cell rate of each source and the long propagation (Wide Area Network) link. After a short transient period, all sources contending for bandwidth are allocated the max-min fair rates. There is an acceptable rate oscillation seen in Figure 3.10 a) to allow the queue to reach the designed level in a long propagation link. However, the steady state behavior is good. The switch queue is conducted to the designed level as shown in Figure 3.10 b). Both links are fully utilized at steady state as shown in Figure 3.10 c). Figure 3.10 d) illustrates that the fairness issue is achieved although the initial cell rates of source 16 and 17 are different.

Figure 3.11 illustrates the results for ACR retention (WAN) scenario.



a) b) c) d)

Figure 3.11: Results for ACR retention (WAN) scenario

In the scenario, the larger number of traversing sessions and long propagation link has been selected to demonstrate the scalability of the FRACA to more VCs and long delay link, as well as to aggravate the problem of ACR retention. Initially, all sources are retaining their cell rates at 10 Mbps then the bottlenecked link could not be utilized as seen in Figure 3.11 c). Consequently, all sources are requested to send more traffic in order to utilize the link. This is corresponding to the continually growth of the allocated rate (until reaching the Peak Cell Rate). After 100 msec, all sources suddenly start sending at their full allocations (then the aggregate traffic is five times larger than the available bandwidth). This causes the switch queue length rapidly grows at 100 msec. The FRACA rapidly detects the overload and gives the appropriate feedback asking all sources to decrease their rates. At this moment, the allocated rate is a little lower than the max-min rate in order to allow the switch to drain the queue. After a short period of draining, the allocated rate converges to the max-min fair rate while the link is fully utilized.

The VBR traffic pattern and the results for 2 source plus VBR (WAN) scenario are shown in Figure 3.12 and 3.13, 2 msec an 10 msec VBR on-off period respectively.

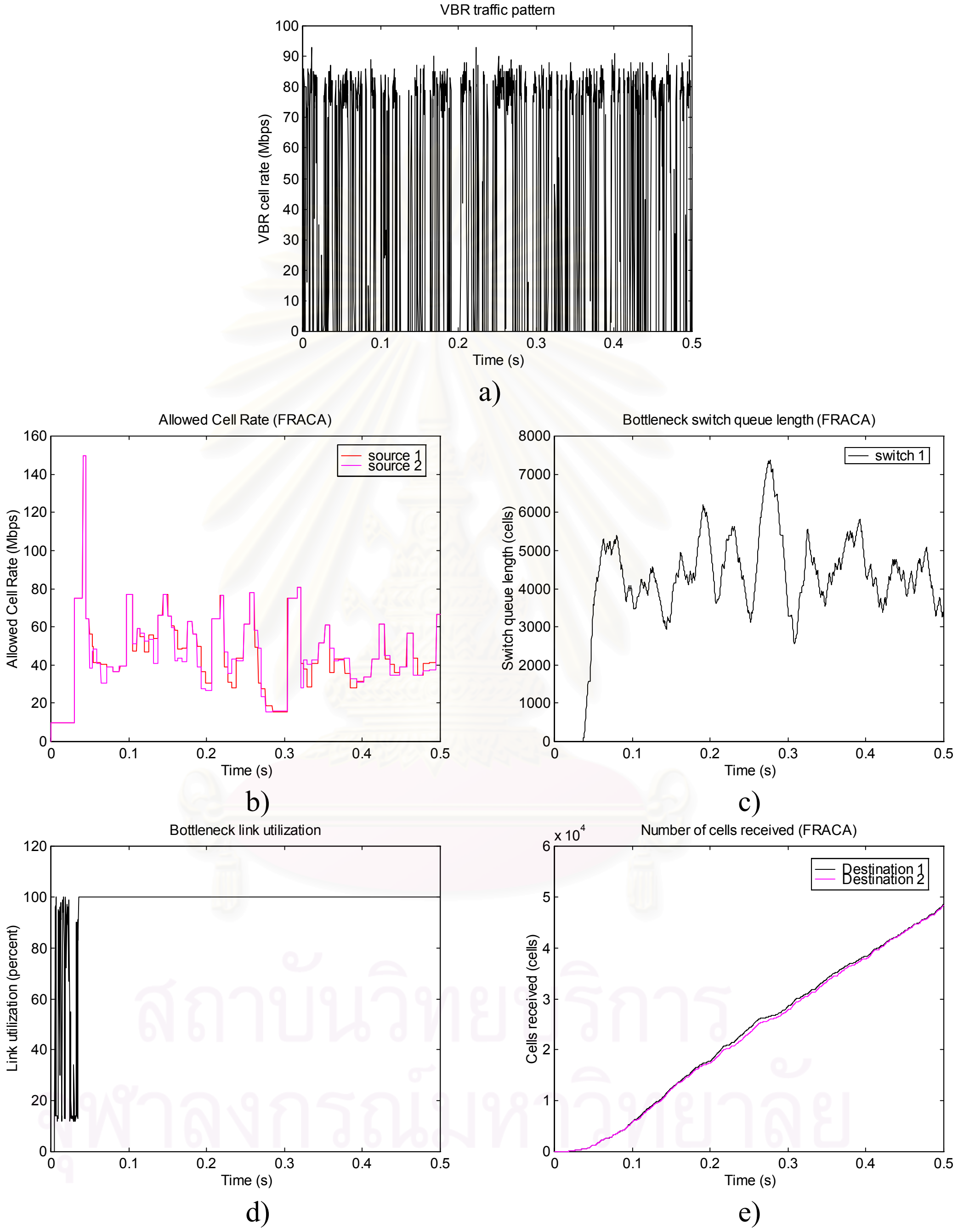
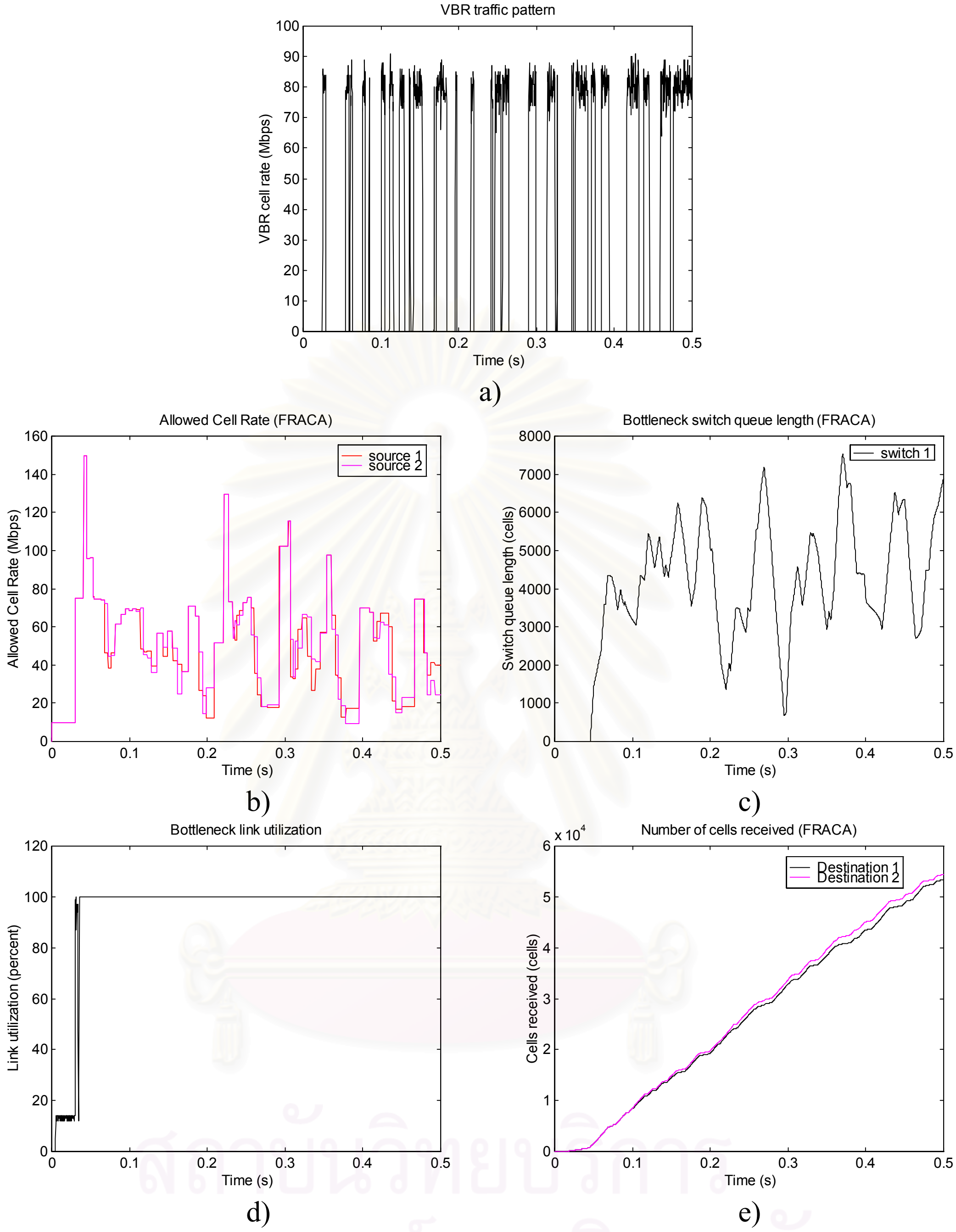


Figure 3.12: Results for 2 Source plus VBR (2 msec on-off period) WAN scenario



a)

b)

c)

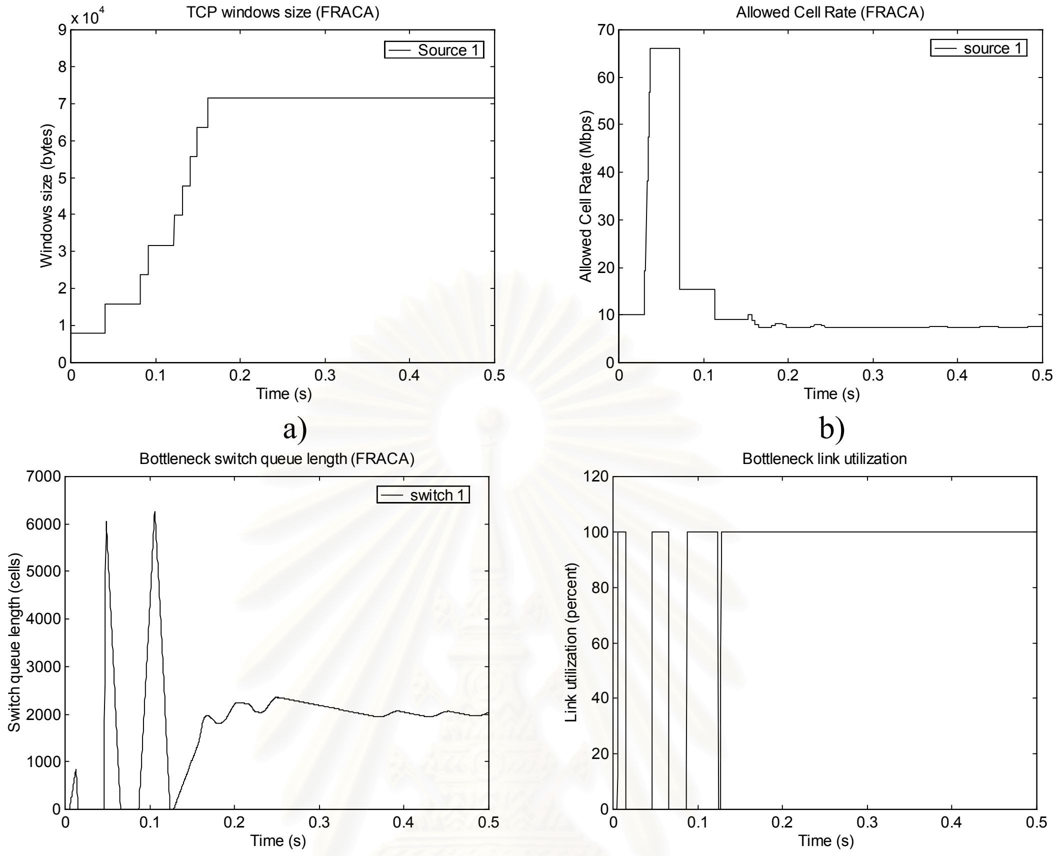
d)

e)

Figure 3.13: Results for 2 Sources plus VBR (10 msec on-off period) WAN scenario

From Figure 3.12, it is obvious that the FRACA rapidly detects the change in the available ABR capacity and gives the appropriate feedback to all sources. When the VBR traffic is active, the ABR sources rapidly reduce their rates. The spikes in the switch queue in Figure 3.12 c) and 3.13 c) reflect the long feedback delay. This is because the queue will drain continuously when VBR traffics immediately change from on-phase to off-phase. As a result, there is no need to wait for BRM cells that will take time to inform the ABR sources. Consequently, the link is fully utilized all the time as shown in Figure 3.12 d) and 3.13 d). The number of cells received in Figure 3.12 d) and 3.13 d) are approximately linear implying that the FRACA averages the available ABR capacity and allocates the max-min fair rate. Generally, Figure 3.12 and 3.13 also illustrate that the more bursty traffic results in the larger amount of buffer occupancy in the switch. However, the switch queue level is kept under control by the FRACA.

The results for 20 TCP over ABR (WAN) scenario are shown in Figure 3.14.



a) b)

c) d)

Figure 3.14: Results for 20 TCP over ABR (WAN) scenario

Figure 3.14 a) shows that during exponential phase of TCP, the aggregated traffic is limited by the rate control of the FRACA. Consequently, the TCP windows size is not exactly exponential. In other words, sometimes the windows size is held for a while if the aggregated traffic is restricted by the rate control. Hence, the sources gradually send the traffic according to the allocated rate. After receiving the ACK that takes a longer period, the sources then increase their windows size. The behavior implies that the rate control policy actually limits the growth of the windows size of TCP applications. During this period, TCP applications act as the on-off sources model with non-uniform on-off period. At steady state, the allowed cell rate converges to the max-min fair rate with full link utilization while the switch queue is kept under control.

Figure 3.15 illustrates the results for the 10 TCP plus VBR over E1 link configuration.

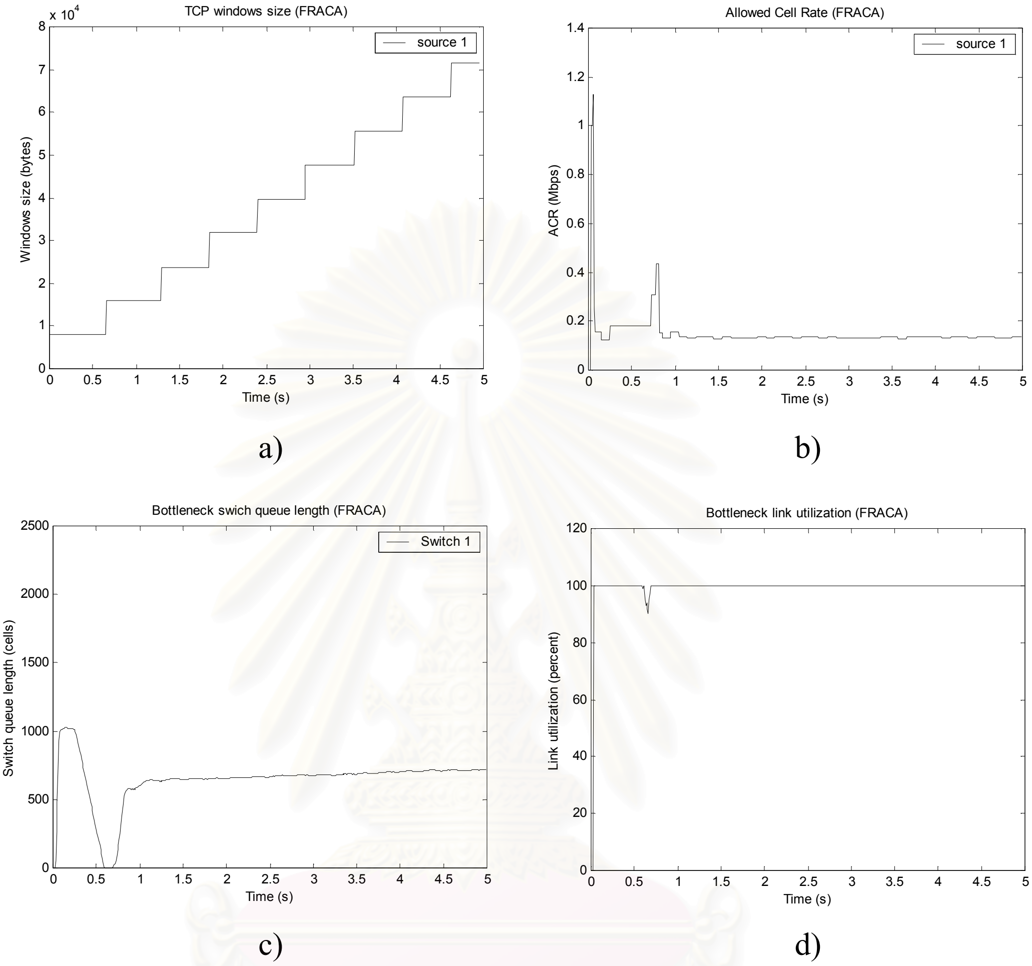


Figure 3.15: Results for 10 TCP plus VBR over E1 link (WAN) scenario

Results in Figure 3.15 indicate that the proposed FRACA algorithm also work well with a low-speed line, for example E1 link, environment. The algorithm averages the aggregate traffics and allocates the max-min fair rates to all ABR sessions in spite of working with the bursty VBR traffic. Consequently, the switch queue length is kept under control.

3.2.9 Performance summation of the FRACA algorithm

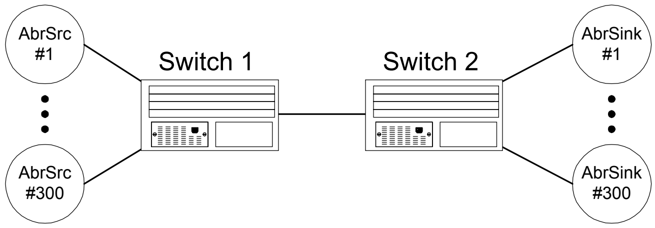
This section has already examined the performance of the proposed FRACA algorithm based on various network configurations. The algorithm entails that the switches periodically monitor their load on each link, determine the available ABR capacity and the load factor in order to allocate the fair max-min rates for all traversing ABR sessions while keeping the switch queue length under control. Various network scenario results indicates that the FRACA algorithm has a fast transient response, achieves full link utilization with short delay and scalable to the variation of source traffics and variable ABR capacity.

3.3 Performance comparison with the ERICA+ and E-FMMRA

In this section, the performance of FRACA is evaluated for 300 ABR sources, GFC-2, staggered TCP and 10 TCP plus VBR scenario to show the enhanced performance, compared with the ERICA+ and E-FMMRA algorithm. For all scenarios, the Allowed Cell Rate (ACR) and switch queue graphs are the metrics used to evaluate the performance of all rate allocation algorithms. A link capacity is fully utilized if the switch queue is kept non-zero. For ERICA+ algorithm, both averaging interval and target queuing delay are fixed set at 5 msec [4]. Setting 5 msec queuing delay corresponds to 1766 ATM cells in 149.76 Mbps link. For E-FMMRA algorithm, the averaging interval is set at the time to receive 100 cells, the low and high threshold of switch buffer occupancy is set at 50 and 1000 cells, respectively [2, 3].

3.3.1 Effects of a large number of traversing ABR sessions

The simulated scenario and parameter setting shown in Figure 3.16 is 300 ABR source configuration. Unlike other researches, we study the effects of a large number of traversing ABR sessions to the performance of our proposed rate allocation algorithm. For robustness point of view, a stable rate allocation algorithm should tolerate to the number of large ABR sessions traversing the link. Consequently, the rate allocation algorithms those work well with a limit number of traversing ABR sessions may not function efficiently in the real working environments.



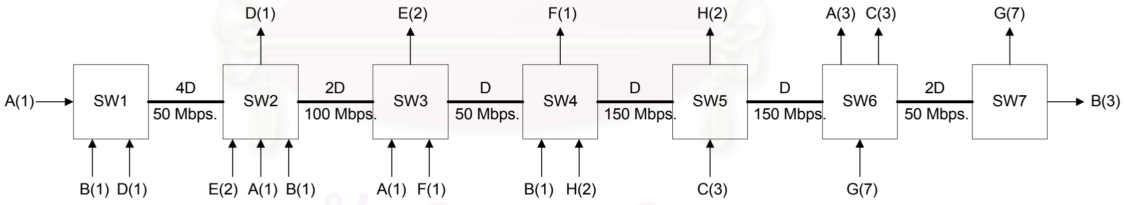
Parameter setting:

- 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 3.16: 300 ABR source configuration

3.3.2 Effects of a complex network scenario

The next simulated configuration is Generic Fairness Configuration (GFC)-2 that is the well-known configuration in ATM Forum [8] for evaluating fairness and robustness issues of the rate allocation algorithms. The configuration consists of both upstream and downstream bottlenecked-link, and parking-lot scenario. The configuration and parameter setting are shown in Figure 3.17.



Note: Entry/exit links of length D, speed 150 Mbps.

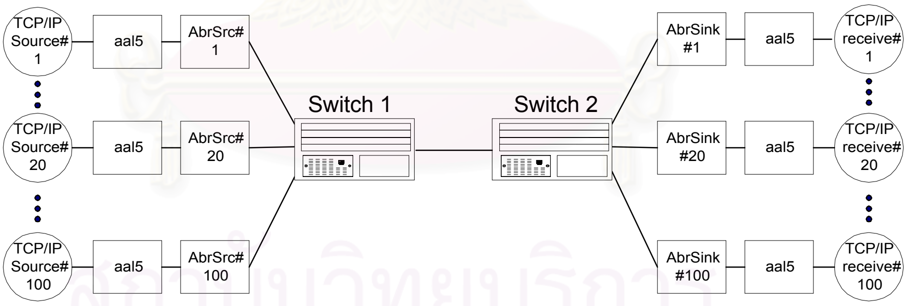
Parameter setting:

- All sources are persistent, PCR = 150.0 Mbps., MCR = 0.0 Mbps., ICR = 5.0 Mbps.
- Propagation delay D = 2.5 msec.
- Output switch buffer size = 10,000 cells.

Figure 3.17: Generic Fairness Configuration (GFC)-2 configuration

3.3.3 Effects of staggered TCP sessions

The next simulated scenario is staggered TCP configuration. When a new TCP session starts sending data, the file transfer data is bottlenecked by the TCP congestion window size (not by ABR control loop). In this state, a TCP session is said to be “window-limited” [15, 23]. On the other hand, TCP congestion window size exponentially grows until it is greater than the source’s sending rate. Then the file transfer rate will be limited by the ABR source rate, not by TCP congestion window size. In this case, a TCP session is said to be “rate-limited”. Hence during the slow start phase, TCP sessions perform as on-off sessions with non-uniform on-off period. A stable congestion avoidance algorithm should not be affected by the aggregated traffic characteristic. The configuration and parameter setting are shown in Figure 3.18. Total 56 bytes of headers and trailers (TCP, IP, RFC2225 and AAL5 layer) are added to every TCP segment. Then every MSS of 8192 bytes becomes 8248 bytes of payload for transmission over ABR service in ATM network. Each payload requires 172 ATM cells of 48 data bytes each. For ABR service category, a FRM cell is sent after sending Nrm-1 (31) data cells. Hence, the maximum throughput of TCP over ATM in this scenario is $\frac{8192}{172\times 53}\times\frac{31}{32}$ = 87.06% of the target ABR capacity.



Parameter setting:

- TCP/IP : TCP sources MSS = 8 kbytes, TCP receiver window size = 64 kbytes.

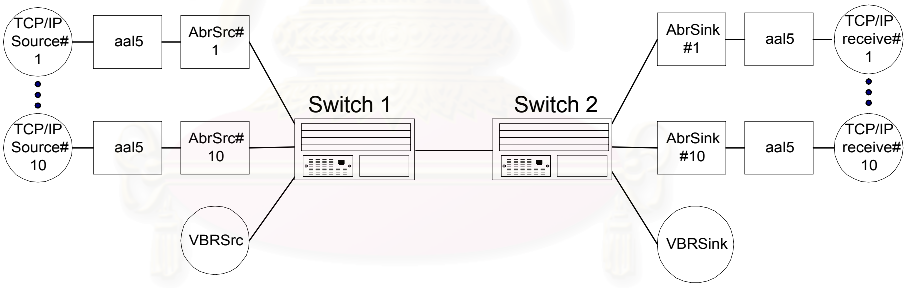
  TCP/IP sources 1-20 turn-on at t = 0 sec., sources 21-100 sequentially turn-on during t = 0.25 to 0.75 sec. For example, source 21 turns on at t = 0.25625 sec., source 22 at t = 0.2625 sec., source 23 at t = 0.26875 sec., …, source 100 at t = 0.75 sec.

- All ABR sources PCR = Link capacity = 149.76 Mbps., MCR = 0.0 Mbps., ICR = 5.0 Mbps.
- All link propagation delay = 500 μsec.

Figure 3.18: Staggered TCP configuration

3.3.4 Effects of TCP sessions with background VBR traffic

The last simulated scenario is 10 TCP sources plus VBR configuration. Unlike other researches, this dissertation presents all rate allocation algorithms that are evaluated under highly bursty on-off VBR source to emulate the worst case of real working environment. The on-off time of VBR traffic is 5 msec and aggregate VBR traffic is 120 Mbps (80% of link capacity) during on period. Others researches [2, 4, 23] used VBR traffic with low variation of aggregate traffic rate which they could not draw conclusion that the designed congestion avoidance algorithm will converge according to max-min rate allocation, for the worst case. The configuration and parameter setting are shown in Figure 3.19.



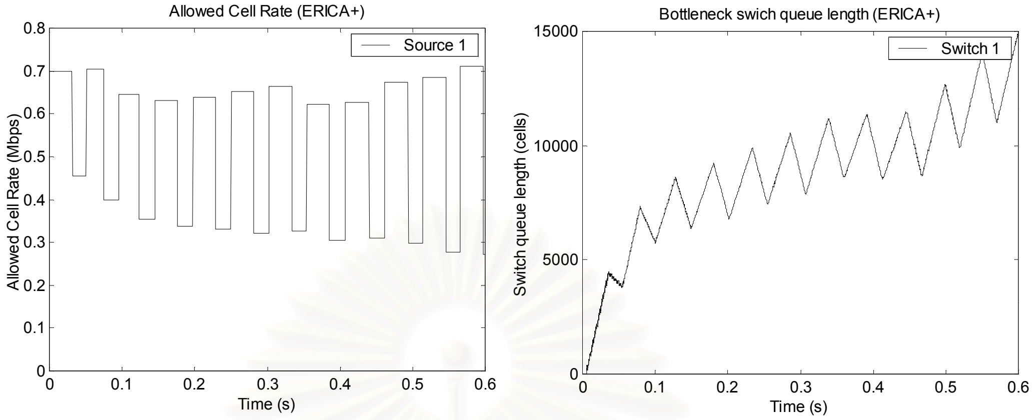
Parameter setting:

- TCP/IP : TCP sources MSS = 8 kbytes, TCP receiver window size = 64 kbytes.
- All ABR sources PCR = Link capacity = 149.76 Mbps., MCR = 0.0 Mbps., ICR = 5.0 Mbps.
- VBR sources are on-off sources, i.e. on-off time is 5 msec., aggregate VBR traffic is 120 Mbps during on period.
- All link propagation delay = 5 msec.

Figure 3.19: 10 TCP sources plus VBR configuration

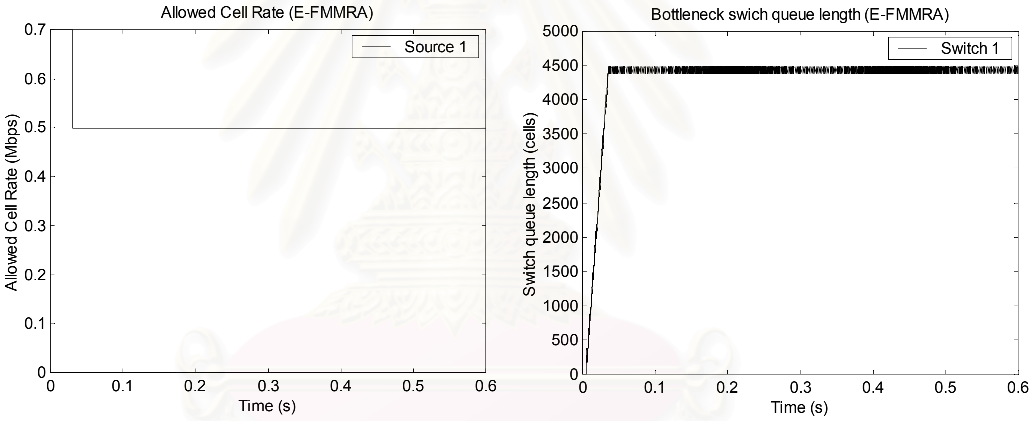
3.3.5 Performance results

This section gives the performance results of the FRACA algorithm compared with the ERICA+ and E-FMMRA under the scenarios mentioned above.



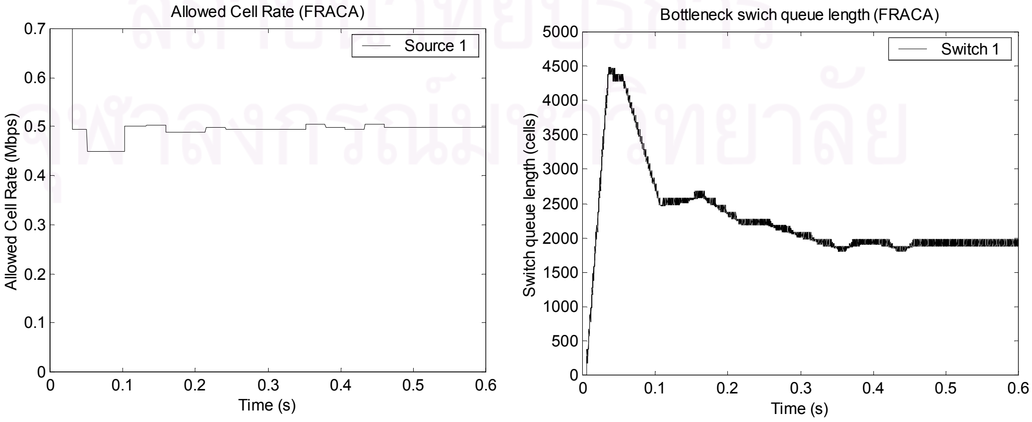
(a) Allowed Cell Rate (b) Switch queue length

Figure 3.20: Results for 300 ABR source configuration (ERICA+)



(a) Allowed Cell Rate (b) Switch queue length

Figure 3.21: Results for 300 ABR source configuration (E-FMMRA)

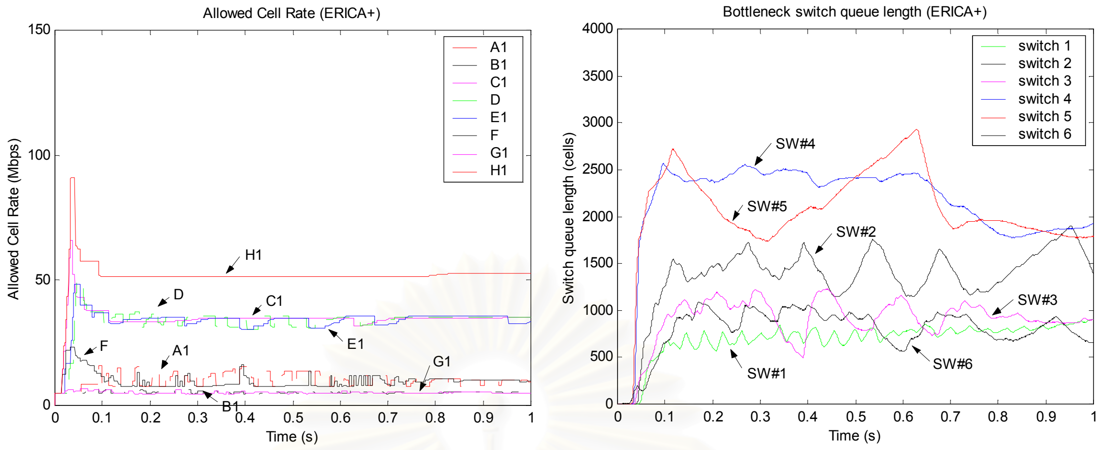


(a) Allowed Cell Rate (b) Switch queue length

Figure 3.22: Results for 300 ABR source configuration (FRACA)

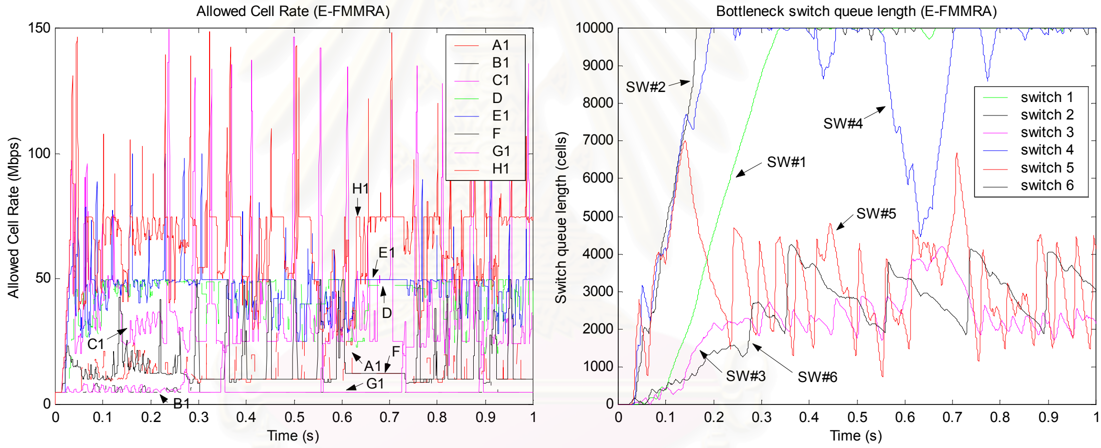
This configuration scenario is used to present the effects of a large number of traversing sessions on congestion avoidance algorithms. The max-min fair rate allocation for all sessions is 149.76/300 ≈ 0.5 Mbps. From simulation results, the ERICA+ faces the severe problem regarding the divergence of sources ACR (see Figure 3.20 a). It is because the algorithm receives all sources BRM cells during the different averaging interval and it uses per average interval computing [7]. As a result, the switch queue length grows unbound in spite of the use of the hyperbolic queue control function. This problem could be solved with the algorithm that runs common weighted average ER. Although allocated rates by E-FMMRA converges to the max-min fair rates but the switch queue length could not be conducted to the designed level (remember that QT and DQT is set at 50 and 1000, respectively). The reason for this is cited in Section 2. Only allocated rates by FRACA algorithm converge to the max-min fair rate and the switch queue length is conducted to the desired level regardless of number of traversing ABR sessions (up to 300 sessions).

Figure 3.23, 3.24 and 3.25 shows the results of GFC-2 configuration for the ERICA+, EFMMRA and FRACA, respectively.



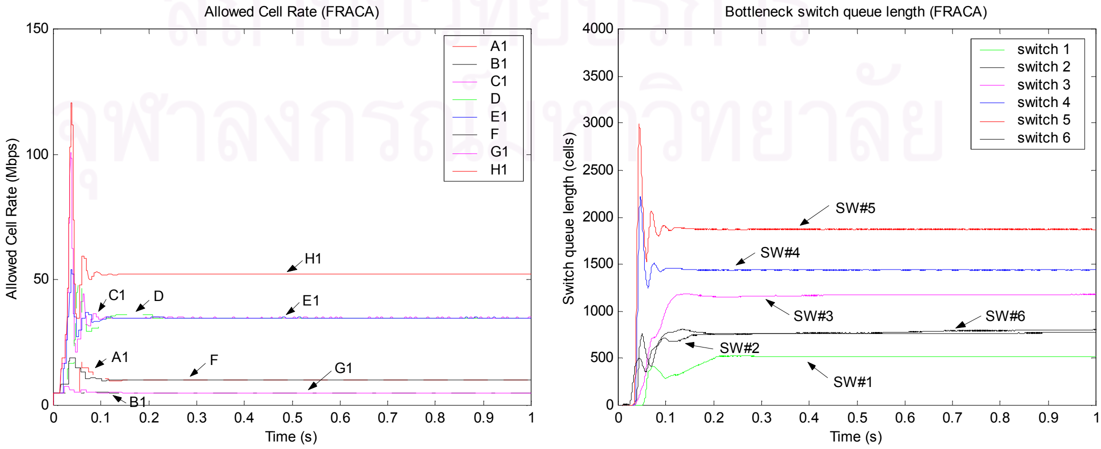
(a) Allowed Cell Rate (b) Switch queue length

Figure 3.23: Results for GFC-2 configuration (ERICA+)



(a) Allowed Cell Rate (b) Switch queue length

Figure 3.24: Results for GFC-2 configuration (E-FMMRA)



(a) Allowed Cell Rate (b) Switch queue length

Figure 3.25: Results for GFC-2 configuration (FRACA)

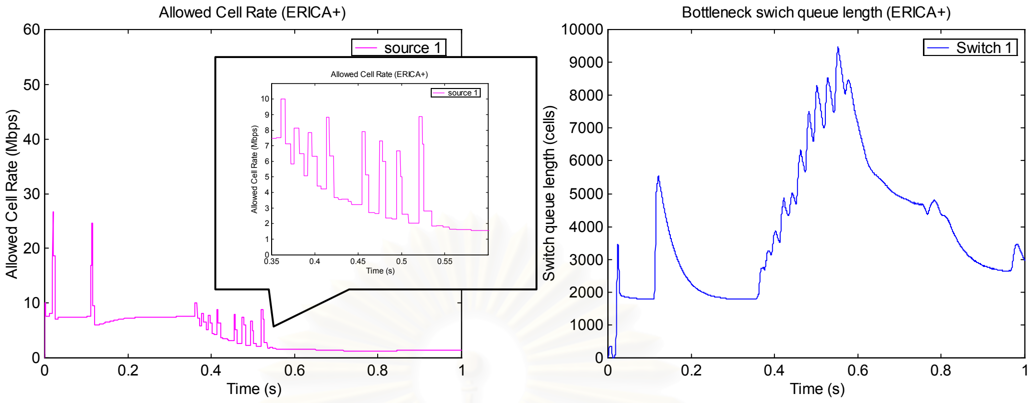
The followings are the expected allocation rates for each session according to the max-min fairness.

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

Discussion

The allocated rates by E-FMMRA algorithm diverge from the max-min rates (Figure 3.24 a) due to the effects from multi-link rate and parking-lot scenario. The algorithm faces the severe cell loss problem in the switches. For ERICA+, the convergence time is very long although the allocated rates converge to the max-min rates (Figure 3.23 a). In addition, the queue length of each switch cannot be conducted to the desired level (Figure 3.23 b) but it oscillates around the desired value. This oscillation is due to the effect of overload condition in working region of the algorithm (remember that at working region $1 < Z < 1+\delta$). The switch queue length indicates that the allocated rates are not exactly the max-min rates. For FRACA algorithm, all allocated rates converge to the expected rates allocation very fast, compared to the other algorithms, with constant controllable queue levels at steady state (Figure 3.25 a, b). This implies that the allocated rates are the max-min fair rates.

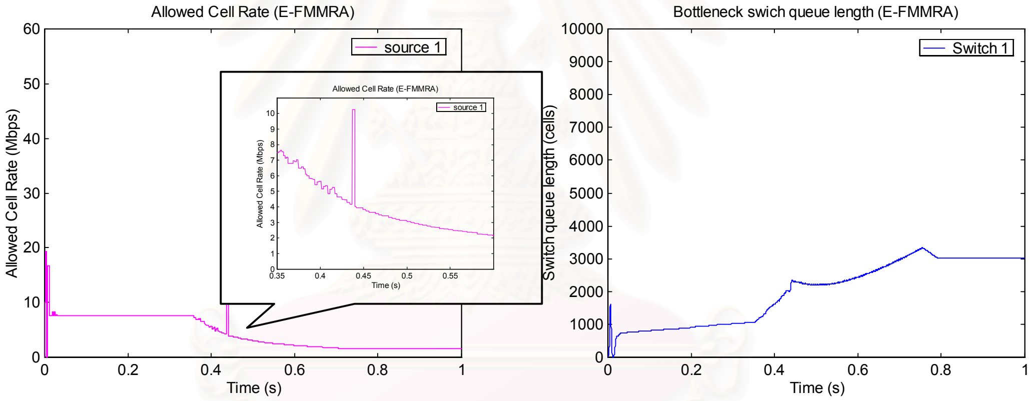
The results of staggered TCP configuration for the ERICA+, EFMMRA and FRACA are shown in Figure 3.26, 3.27 and 3.28, accordingly.



(a) Allowed Cell Rate

(b) Switch queue length

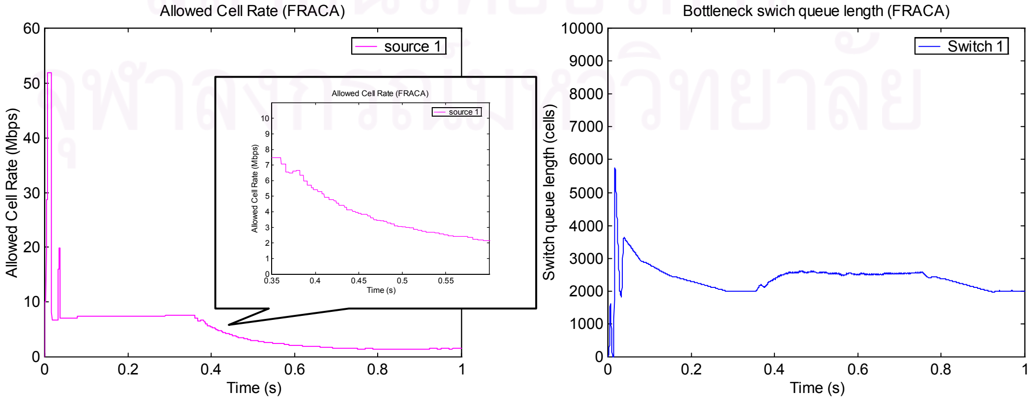
Figure 3.26: Results for staggered TCP configuration (ERICA+)



(a) Allowed Cell Rate

(b) Switch queue length

Figure 3.27: Results for staggered TCP configuration (E-FMMRA)



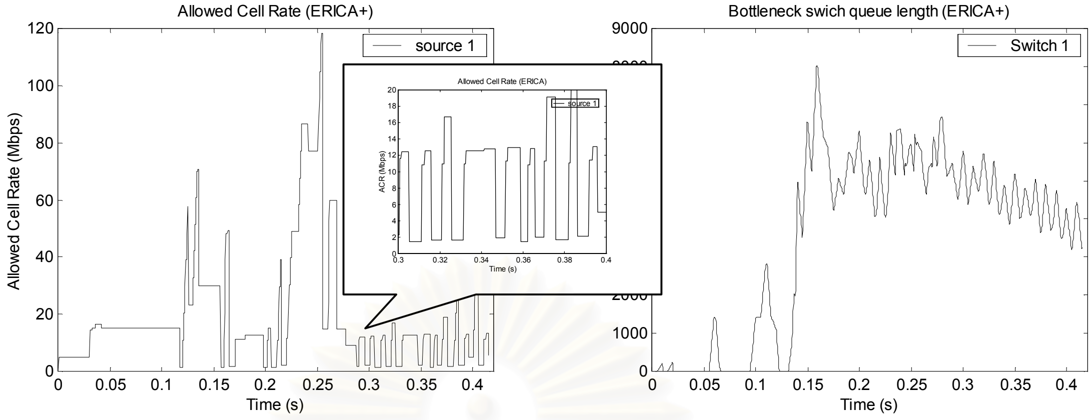
(a) Allowed Cell Rate

(b) Switch queue length

Figure 3.28: Results for staggered TCP configuration (FRACA)

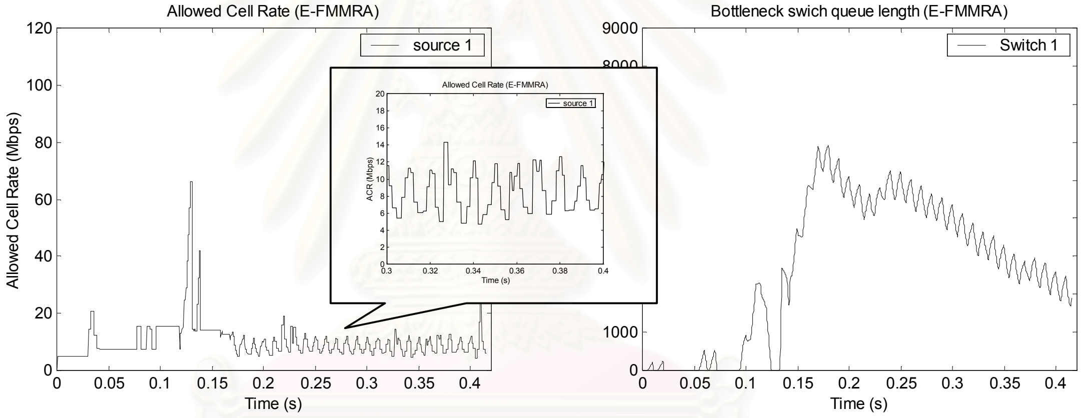
This scenario is presented to illustrate the effects of slow start (exponential) phase of TCP application on congestion avoidance algorithms. The TCP sources number 21 through 100 sequentially turn on from time t = 0.25 to 0.75 sec. Therefore, they periodically turn on every (0.75 – 0.25)/80 = 6.25 msec. Moreover, from the MSS and window-sized TCP receiver in the configuration, it will take at least the time of 3 round-trips for TCP source in an exponential phase to become a greedy source. The time of 3 round-trips is equal to 3 × 2 × (500 μsec × 3) = 9 msec., which is greater than the time that a new source turns on. As the result, the configuration may be assumed to be a mixture of persistent and on-off sources during time t = 0.25 ~ 0.75 sec. After the window size of each TCP source is increased to its maximum value, the source rate is then not limited by the window size but controlled by the ABR loop. All TCP connections perform as persistent sources. Therefore, the allocated rates by all algorithms should be the steady state max-min values after t=0.75 sec. From simulation results, source 1 ACR of the FRACA algorithm (Figure 3.28 a) rapidly grows during transient phase because of faster transient response, compared to others, causing a slightly greater buffer occupancy in the switch (Figure 3.28 b). However, the queue level is conducted to the desired level when the algorithm detects overload condition. In addition, during the periodic turn-on period of staggered TCP sources, the ACR of FRACA shows good transient response. This causes the switch queue level controllable compared to ERICA+ and E-FMMRA. For ERICA+, the switch queue level continually grows (Figure 3.26 b) due to the error of allocated rates. For E-FMMRA, the switch queue length could not be conducted to the desired level (Figure 3.27 b) at steady state. Therefore, the performance of FRACA algorithm is tolerant to a TCP slow start phase and staggered TCP applications.

Figure 3.29, 3.30 and 3.31 shows the results of 10 TCP sources plus VBR configuration for the ERICA+, EFMMRA and FRACA, respectively.



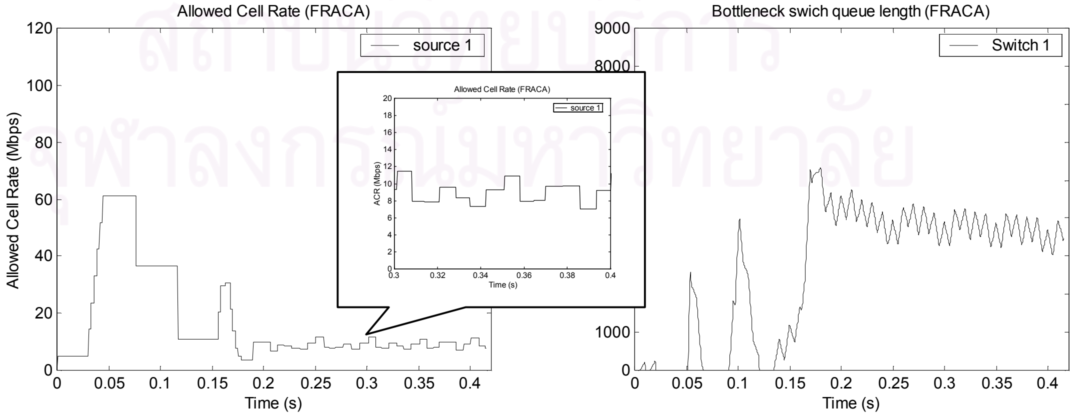
(a) Allowed Cell Rate (b) Switch queue length

Figure 3.29: Results for 10 TCP sources plus VBR configuration (ERICA+)



(a) Allowed Cell Rate (b) Switch queue length

Figure 3.30: Results for 10 TCP sources plus VBR configuration (E-FMMRA)



(a) Allowed Cell Rate (b) Switch queue length

Figure 3.31: Results for 10 TCP sources plus VBR configuration (FRACA)

The expected explicit rate for all TCP sessions is approximate 9 Mbps. $\left[\frac{149.76-120/2}{10}\right]$. The rate allocation algorithm should be able to average the aggregate traffic and compute this fair rate for all ABR sessions despite of working with highly bursty VBR traffic. The rate allocation algorithm that uses time-based averaging interval (ERICA+) or too short count-based averaging interval (E-FMMRA) cannot accurately average the aggregate traffic resulting in the oscillation of the allocated rates. The solution for designing a robust rate allocation algorithm is to use appropriate length of count-based averaging interval in order to correctly average aggregate traffic while maintaining fast transient response according to the design concept of the FRACA algorithm. Comparison of Figure 3.29, 3.30 and 3.31, it is obvious that the FRACA algorithm gives the better performance than that of ERICA+ and E-FMMRA in terms of ACR. Similar to the results for staggered TCP configuration, source 1 ACR of the FRACA rapidly grows during start up period causing a slightly greater switch queue length. However, it is kept under control after the switch detects the overload condition.

3.3.6 Summary of performance comparison of the FRACA algorithm with the ERICA+ and E-FMMRA

As the goals for the design of a rate allocation algorithm, the FRACA has been shown to be convergent in all tested scenarios. In addition, the performance is tolerant to network conditions such as number of traversing ABR sessions, system propagation delay, network scenario and aggregate traffic characteristics. The designed algorithm has been shown to have a better transient and steady state response compared with the ERICA+ and E-FMMRA. Table 3.2 presents more extensive performance comparison for ERICA+, E-FMMRA and FRACA algorithm.

Table 3.2: Comparison of explicit rate allocation algorithms

ERICA+	- The ERICA+ may not converge to the max-min fairness in a large number of traversing ABR sessions and staggered TCP connections. Consequently, the switch queue may not be properly controlled. - The algorithm is sensitive to the CCR error due to the use of CCR field in a FRM cell in the rate allocation process. - The algorithm may not accurately average the aggregate traffic, especially when working with bursty VBR traffic, due to the time-based averaging interval and the per-averaging interval computation of the algorithm.
E-FMMRA	- The E-FMMRA may not converge to the max-min fairness in a parking-lot or multi-link rate configurations especially in the WAN environments. Therefore, the switch queue may not be properly controlled. - The algorithm is insensitive to the CCR error because it does not use the CCR value. - The algorithm may not accurately average the aggregate traffic due to the recommended short count-based averaging interval.
FRACA	- The FRACA converges according to the max-min fairness in all evaluated configurations. As the result, a switch queue is properly controlled. - The algorithm is insensitive to the CCR error because it does not use the CCR value. - The algorithm accurately averages the aggregate traffic due to the appropriate length of count-based averaging interval. However, the algorithm may use larger buffer size than the algorithm that uses shorter averaging interval; for example, the E-FMMRA.

For robustness and scalability issues, only the FRACA algorithm achieves the design goals with our target environments. However, there are some limitations of the FRACA algorithm. Firstly, since the per-connection accounting is employed in the rate allocation process as well as the ERICA+ and E-FMMRA, the memory size needed for the algorithm is proportional to the number of multiplexed connections, which renders the algorithm hard to be implemented in a large-scale network. Secondly, in order to average the aggregated traffic more accurately, the longer averaging interval is used by FRACA, compared to the E-FMMRA. With the existence of bursty VBR traffic in the network, the FRACA algorithm may allocate the max-min rates slower than the E-FMMRA. Consequently, the FRACA algorithm may need larger buffer size in some scenarios. In this respect, if the switch buffer is too limited, then the averaging interval length of the FRACA algorithm should be reduced from the recommended value in order to provide faster responses. However, a too short averaging interval length may causes allocated rates to be hard to converge to the max-min fair rates, as a result of the inaccurate traffic averaging.

3.4 The FRACA Convergence Analysis

This section presents a mathematical proof of the convergence of the FRACA algorithm. It is assumed that all sources are persistent and have zero MCR requirement. Moreover, the numbers of sessions traversing links do not change during the convergence period.

Definitions and notations

- A group of *bottlenecked sessions* $(\Im)$ includes all the sessions that can utilize higher rate allocation but they are bottlenecked by the assigned rate.
- A group of *non-bottlenecked sessions* $(\bar{\Im})$ includes all the sessions that are bottlenecked elsewhere. Thus they cannot utilize the AR.
- Total usage by non-bottlenecked sessions $(C_{\bar{\Im}}) = \sum_{k\in\bar{\Im}} ER_k$
- λ is the number of bottlenecked sessions.
- $\bar{\lambda}$ is the number of non-bottlenecked sessions.
- ℓ is the set of uni-directional links in the network.
- r_j is the max-min rate allocation of level-j bottlenecked grouping.
- ℓ_{b_j} is a level-j bottlenecked link, which level-j bottlenecked grouping traverses.
- $n(A)$ denotes number of elements in set A.

- Each session has an associated set of links representing its traversed path $\{\ell_1, \ell_2, ..., \ell_n\} \subseteq \ell$
- t_a is the averaging interval time.
- $t_{\max}$ is the longest Round Trip Time for all sessions.
- $\lceil n \rceil$ is the integer value that is not less than *n*.

In this section, the theorems are developed based on the definition of max-min fair rate presented in [15]. Let R_i be the sum of all allocated rates on link i, i.e.,

$$R_i = \sum_{\substack{\text{all sessions k} \\ \text{traversing link i}}} r_{k_i} \qquad (3\text{-}1)$$

where r_{k_i} is an allocated rate of session k on link i. Further, let $\Re_i$ denote the capacity of link i. So we have the following constraints on the allocated rates:

$$r_{k_i} \geq 0 \qquad \text{for all sessions traversing link i}$$

$$R_i \leq \Re_i \qquad \text{for all link } (\ell_i \in \ell) \qquad (3\text{-}2)$$

An allocated rate satisfying the above constraints is said to be *feasible*. In order to prove the convergence of FRACA algorithm, the following definitions are fist given.

Definition 1

An allocated rate is said to be *max-min fair* if it is feasible. And for each link $(\ell_i \in \ell)$, any allocated rate of a session traversing link i (r_{k_i}) cannot be increased while maintaining feasibility without decreasing others session $(r_{k'_i})$ for which $r_{k'_i} \leq r_{k_i}$. ■

Definition 2

A link i is a *bottleneck link* with respect to the allocated rate of session k traversing the link (r_{k_i}) if $R_i = \Re_i$ and $r_{k'_i} \leq r_{k_i}$ for other session (k') traversing link i. In other words, a link i is bottlenecked for a session k if the link bandwidth is fully utilized and session k has the largest allocated rate among all sessions traversing link i. ■

Definition 3

A rate allocation algorithm is said to be *convergent* if the algorithm allocates the max-min fair rates to all bottlenecked sessions regardless of sources' initial rate.

■

The followings present the convergence analysis of the single link scenario and multiple-link scenario.

3.4.1 Single link scenario

Consider a network link with capacity $\Re$ cells/second, supporting a set of N traversing sessions. The sessions are classified into the appropriate groupings, bottlenecked $(\Im)$ and non-bottlenecked $(\bar{\Im})$, accordingly.

From the definition, a session is said to be a non-bottlenecked session if the allocated rate is lower than the fairshare. For example, the sessions are bottlenecked by their PCR. That is,

$$ER_k < \frac{\Re}{N} \qquad \ldots \ \forall k \in \bar{\Im} \qquad (3\text{-}3)$$

Summing the above over all non-bottlenecked sessions gives the total usage by non-bottlenecked sessions:

$$C_{\bar{\Im}} = \sum_{k \in \bar{\Im}} ER_k < \bar{\lambda} \cdot \frac{\Re}{N} \qquad (3\text{-}4)$$

The above expression implies that the aggregate bandwidth usage by non-bottlenecked sessions cannot exceed their proportional bandwidth share at the link. In the scenario, the subsequent convergence argument is proved as follows,

Lemma 1

A session is bottlenecked if the computed ER is greater than or equal to the max-min fair rate for this session, i.e.,

$$ER_k \geq AR = \frac{\Re - C_{\bar{\Im}}}{\lambda} \qquad \ldots \ \forall k \in \Im \qquad (3\text{-}5)$$

Proof

By definition, the sessions are bottlenecked if they can utilize higher rate allocation until the link is fully utilized. The link, therefore, becomes bottleneck for these sessions. It is clear that bottlenecked sessions should equally share the bandwidth left from non-bottleneck sessions. The allocated rate is max-min fair according to definition 1. ■

Using Lemma 1, the FRACA algorithm converges according to the following theorem.

Theorem 1

For single link scenario, if all sessions are persistent, the minimum time that the FRACA algorithm will converge according to the max-min fairness is $\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a$.

Proof

Let us separate the proof into two parts:

Part A: All sources are not bottlenecked by their PCRs. In other words, every source's PCR is greater than the fairshare. All sessions become bottlenecked sessions within the longest Round Trip Time $(t_{\max})$.

In this case, all sources are expected to be marked as bottlenecked sessions and to be assigned the fairshare as feedback explicit rate. Upon the receipt of the BRM cell, the switch allocates at least the fairshare (AR) to all sources according to:

$$AR = \frac{\Re - C_{\bar{\Im}}}{\lambda} = \frac{\Re}{N} \tag{3-6}$$

Remember that $C_{\bar{\Im}} = 0$ and $\lambda = N$. The link becomes bottlenecked with respect to all sessions. Hence, all sessions become bottlenecked sessions within the longest RTT, according to Lemma 1.

Part B: Some sources are bottlenecked by their PCRs. In other words, some sources' PCR is lower than the fairshare. The remaining sessions become bottlenecked sessions within $\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a$.

During the first cycle, before receiving the first BRM cell, all sessions are firstly classified as bottlenecked sessions. After all sources receive their first BRM cells, the algorithm will classify non-bottlenecked sessions from all sessions and their corresponding bandwidth are recorded. The AR is recomputed as

$$AR = \frac{\Re - C_{\overline{\Im}}}{\lambda} \qquad (3\text{-}7)$$

The algorithm will assign at least this AR to the rest of sessions during the next averaging interval. According to definition 2, the link becomes bottlenecked with respect to these sessions. Hence, the sessions become bottlenecked according to Lemma 1. ■

By definition 3, the FRACA algorithm is proved to be convergent within $\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a$, in a single link scenario.

3.4.1.1 Conformity Analysis

This section presents the conformity analysis between the analytical proof and the simulation result, in a single link scenario. Let us use the simulation result of the 300 ABR source configuration in Figure 3.22 a) to evaluate the consistence. For clarification, let see the result shown in Figure 3.32 (the same as Figure 3.22).

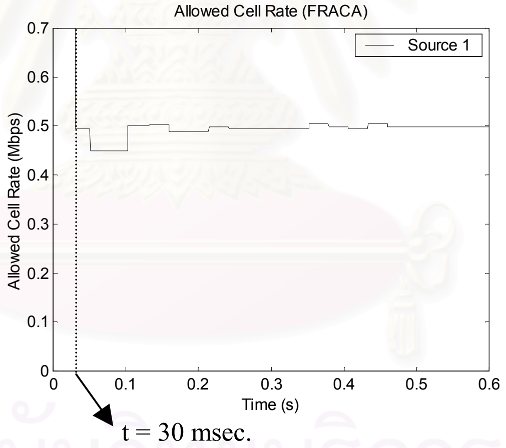


Figure 3.32: ACR for source 1 of 300 ABR source configuration

From the configuration in Figure 3.16, the round trip time for all sources is 6×5 msec. = 30 msec. Consequently, the minimum time that the algorithm will converge is the time when the first averaging interval after t = 30 msec. is expired. We can say approximately 30 msec. From the result in Figure 3.32, the ACR rapidly drop from the ICR that is set at 0.7 Mbps. to the max-min fair rate that is approximate 0.5 Mbps. about t = 30 msec. From the result, in addition, the ACR is a little lower than the max-min fair rate during t = 60 to 100 msec. The reason is to control the switch queue. Then, the simulation result in the case of a single link scenario with persistent sources is consistent with the analytical proof given in Section 3.4.1.

3.4.2 Multiple-link scenario

In multiple-link scenario, we initially consider the link carrying the largest number of sessions. For example, link i maximizes $\lambda_1 + \overline{\lambda_1}$. All sessions traversing the link belong to level-one bottlenecked grouping, denoted by β_1, and have the steady-state max-min fair rate

$$r_1 = \min_{i \in \ell} \left\{ \frac{\Re_i}{\lambda_i + \overline{\lambda_i}} \right\} \qquad \ldots \ \forall \ell_i \in \ell \qquad (3\text{-}8)$$

where λ_i and $\overline{\lambda_i}$ are the numbers of steady state bottlenecked and non-bottlenecked sessions at link i, respectively. Let N_i denotes number of sessions traversing link i. Then $N_i = \lambda_i + \overline{\lambda_i}$. By iterative procedure, the second-most bottlenecked link is derived by removing all level-one bottlenecked sessions and their traversing link(s) from the network. The level-one bottlenecked link in the reduced network is the second-most bottlenecked link. All sessions traversing the second-most bottlenecked link now belong to level-two bottlenecked grouping, denoted by β_2. Thus, it shows that the max-min fair rate for sessions in level-j bottlenecked grouping, β_j (j ≥ 1), can be calculated from:

$$r_j = \min_{i \neq \ell_{b_1}, \ell_{b_2}, \ldots, \ell_{b_{j-1}}} \left\{ \frac{\Re_i - \sum_{m=1}^{j-1} \left[r_m \cdot n\left(\beta_m \cap \left(\Im_i \cup \overline{\Im_i}\right)\right)\right]}{N_i - \sum_{m=1}^{j-1} n\left(\beta_m \cap \left(\Im_i \cup \overline{\Im_i}\right)\right)} \right\} \qquad (3\text{-}9)$$

where $\Im_i$ and $\overline{\Im_i}$ are steady state bottlenecked and non-bottlenecked groupings at link i, respectively. Next, let b$_i$ be the bottlenecked-level for link i. Therefore, b$_i$ - 1 bottlenecked-level become non-bottlenecked sessions at link i by definition. Let $C_{\overline{\Im_i}}$ be the total bandwidth usage by non-bottlenecked sessions at link i, hence

$$C_{\overline{\Im_i}} = \sum_{k \in \overline{\Im_i}} r_k = \sum_{k=1}^{b_i - 1} \left[r_k \cdot n\left(\beta_k \cap \left(\Im_i \cup \overline{\Im_i}\right)\right)\right] \qquad \ldots \ b_i > 1 \qquad (3\text{-}10)$$

Furthermore, let $\overline{\Im_i^j}$ denote level-j non-bottlenecked grouping at link i, 1 < j < b$_i$ - 1. The level-j non-bottlenecked steady state bandwidth usage, $C_{\overline{\Im_i^j}}$, is given by

$$C_{\overline{\Im_i^j}} = \sum_{k \in \left(\overline{\Im_i} \cap \beta_j\right)} r_k = r_j \cdot n\left(\overline{\Im_i} \cap \beta_j\right) = r_j \cdot \overline{\lambda_i^j} \qquad (3\text{-}11)$$

where $\overline{\lambda_i^j}$ is the number of level-j non-bottlenecked sessions traversing link i. Then the relation between (3-10) and (3-11) is

$$C_{\overline{\Im_i}} = \sum_{j=1}^{b_i-1} C_{\overline{\Im_i^j}} \qquad (3\text{-}12)$$

The followings are the subsequent convergent arguments in multiple-link scenario.

Lemma 2

The steady state fair rate for bottlenecked sessions at link i is greater than or equal to the link steady state max-min fair rate.

$$ER_i^k \geq r_{b_i} \qquad \cdots \ \forall k \in \Im_i \qquad (3\text{-}13)$$

Proof

The advertised rate or the max-min fair rate for bottlenecked sessions at link i, i.e. level-b$_i$ bottlenecked grouping, is computed from

$$AR = r_{b_i} = \frac{\Re_i - C_{\overline{\Im_i}}}{\lambda_i} \qquad (3\text{-}14)$$

If all lower level bottlenecked sessions at link i have stabilize at their max-min rate, these sessions are non-bottlenecked at the link by definition, that is $\left[\beta_k \cap \left(\Im_i \cup \overline{\Im_i}\right)\right] \subseteq \overline{\Im_i}$ for $k < b_i$. The remaining sessions should receive r_{b_i} as feedback explicit rate. However, some sessions may be bottlenecked by their PCRs, i.e. $r_{b_i-1} < ER_i^{k'} < r_{b_i}$ for $k' \in \left(\overline{\Im_i} \cap \beta_{b_i}\right)$. Hence, the max-min fair rate for the other sessions is computed as

$$\begin{aligned} ER_i^k &= \frac{\Re_i - \sum_{j=1}^{b_i-1} C_{\overline{\Im_i^j}} - \sum_{k' \in \left(\overline{\Im_i} \cap \beta_{b_i}\right)} ER_i^{k'}}{N_i - \sum_{j=1}^{b_i-1} \overline{\lambda_i^j} - n\left(\beta_{b_i} \cap \left(\Im_i \cup \overline{\Im_i}\right)\right)} \\ &= \frac{\Re_i' - \sum_{k' \in \left(\overline{\Im_i} \cap \beta_{b_i}\right)} ER_i^{k'}}{N_i' - n\left(\beta_{b_i} \cap \left(\Im_i \cup \overline{\Im_i}\right)\right)} \\ &= \frac{\Re_i' - C_{\overline{\Im_i'}}}{N_i' - \overline{\lambda_i'}} = \frac{\Re_i' - C_{\overline{\Im_i'}}}{\lambda_i'} \end{aligned} \qquad (3\text{-}15)$$

Where $\Re_i' = \Re_i - \sum_{j=1}^{b_i-1} C_{\overline{\Im_i^j}}$ is the left over bandwidth from level-1 through level-$b_i - 1$ non-bottlenecked sessions traversing link i.

$N_i' = N_i - \sum_{j=1}^{b_i-1}\overline{\lambda_i^j}$ is the number of sessions left over from level-1 through level-$b_i - 1$ non-bottlenecked sessions traversing link i.

$C_{\overline{\Im_i'}} = \sum_{k'\in\left(\overline{\Im_i}\cap\beta_{b_i}\right)} ER_i^{k'}$ is the total bandwidth usage by level-b_i non-bottlenecked sessions traversing link i that are bottlenecked by their PCRs.

$\overline{\lambda_i'} = n\left(\beta_{b_i} \cap \left(\Im_i \cup \overline{\Im_i}\right)\right)$ is the number of level-b_i non-bottlenecked sessions traversing link i that are bottlenecked by their PCRs.

Then the link i becomes bottlenecked with respect to these sessions. From definition and iterative computation, the computed $ER_i^k(\forall k \in \Im_i)$ is always higher than or equal to, i.e. if $C_{\overline{\Im_i'}} = \overline{\lambda_i'} = 0$, the allocated rate of level-b_i bottlenecked sessions traversing link i. ■

Lemma 3

For multiple-link scenario, if all sources are persistent, the minimum time that the FRACA algorithm will converge according to the max-min fairness is $\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a$.

Proof

This is the case of downstream bottleneck scenario. During backward path of the first BRM cell, the expected non-bottlenecked sessions are sequentially marked as the lower-level bottlenecked grouping at the less congested link. Then the intermediate switches compute and allocate the advertised max-min fair rate (3-15) to all bottlenecked sessions at the end of the averaging interval after the longest RTT.

■

Lemma 4

For multiple-link scenario, if all sources are persistent, the FRACA algorithm will converge according to the max-min fairness within $n < \log_{1+\varepsilon} \frac{\Re_i}{\sum_{j=1}^{b_i} \sum_{k\in\left(\Im_i\cap\beta_j\right)} ER_i^k\left(\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a\right)}$ averaging intervals.

Proof

This is the case of upstream bottleneck scenario. The important task is the way to let the downstream switches know the bottleneck condition in the upstream switches because the rate allocation algorithm works only in backward direction. The load factor, defined as the ratio of ABR input rate to ABR Capacity (the queue control factor is relaxed), is introduced to solve the problem. Let t' denote the time which the FRACA algorithm converges according to max-min fair rate allocation, i.e. $ER_i^k(t') = ER_i^k$ according to (3-15), and let ρ_{i_t} be the load factor of link i during time interval $(t \to t + t_a)$.

Basically in the case of upstream bottleneck, initially, the allocated rate at downstream link cannot fully utilize the link capacity. Thus, the load factor is below one. The allocated rate will be continually increased as the result from equation (3-1) and (3-3). The downstream link is fully utilized if the load factor becomes one $(R_i = \Re_i)$. That is

$$1 = \rho_{i_{t'}} > \rho_{i_{t'-t_a}} > \rho_{i_{t'-2t_a}} > \ldots > \rho_{i_{\left\lceil \frac{t_{\max}}{t_a} \right\rceil t_a}} \qquad (3\text{-}16)$$

Thus the system will converge to the max-min rate allocation. In other words the allocated rates for all bottlenecked sessions are the max-min fair rates (3-15). Equation (3-16) could be rewritten as

$$1 = \rho_{i_{t'}} = \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} ER_i^k(t')}{\Re_i} > \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} ER_i^k(t'-t_a) \Big/ \rho_{i_{t'-t_a}}}{\Re_i} > \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} ER_i^k(t'-2t_a) \Big/ \rho_{i_{t'-t_a}} \cdot \rho_{i_{t'-2t_a}}}{\Re_i} > \ldots > \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} ER_i^k\left(\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a\right) \Big/ \rho_{i_{t'-t_a}} \cdot \rho_{i_{t'-2t_a}} \cdot \ldots \cdot \rho_{i_{\left\lceil \frac{t_{\max}}{t_a} \right\rceil t_a}}}{\Re_i}$$

$$(3\text{-}17)$$

By using the analytic technique given in [4], the allocated rates converge to the max-min fair rates if the load factor $\rho_{t'-t_a}$ is less than $\frac{1}{1+\varepsilon}$ for small positive ε. Hence,

$$\rho_{i_{t'}} > \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} ER_i^k\left(\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a\right) \Big/ \rho_{i_{t'-t_a}} \cdot \rho_{i_{t'-2t_a}} \cdot \ldots \cdot \rho_{i_{\left\lceil \frac{t_{\max}}{t_a} \right\rceil t_a}}}{\Re_i} > \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} ER_i^k\left(\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a\right) \Big/ \left(\frac{1}{1+\varepsilon}\right)^n}{\Re_i} \qquad (3\text{-}18)$$

where *n* is the number of averaging intervals (cycles) to which the FRACA algorithm converges according to the max-min fair rate after receiving the first BRM of the longest RTT session, i.e. $n = \left\lceil \frac{t' - t_{\max}}{t_a} \right\rceil$. From (3-17) and (3-18), it is clear that the FRACA algorithm will converge according to the following constraint

$$\frac{\sum_{j=1}^{b_i}\sum_{k\in(\Im_i\cap\beta_j)} ER_i^k\left(\left\lceil\frac{t_{\max}}{t_a}\right\rceil\cdot t_a\right)}{\left(\frac{1}{1+\varepsilon}\right)^n}<\Re_i$$

Therefore,
$$n<\log_{1+\varepsilon}\frac{\Re_i}{\sum_{j=1}^{b_i}\sum_{k\in(\Im_i\cap\beta_j)} ER_i^k\left(\left\lceil\frac{t_{\max}}{t_a}\right\rceil\cdot t_a\right)} \tag{3-19}$$

The small positive ε implies how close does the actual load factor (ρ) and the ideal value, that is unity, as shown in Figure 3.33. If ε becomes smaller then the algorithm will take more cycles of the averaging interval to achieve the max-min fairness.

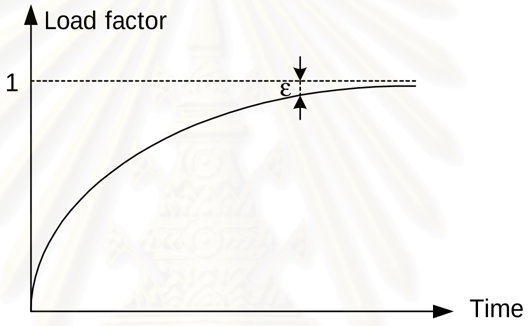


Figure 3.33: Characteristics of the averaging interval in the upstream bottleneck of the multiple-link scenario

Equation (3-19) implies that the FRACA algorithm will converge to the max-min fair rate in finite time. ■

In addition, if we apply equation (3-19) with the single link scenario then *n* becomes zero. That is because the sum of all explicit rate for all traversing sessions is the ABR capacity. Thus, the algorithm will converge to the max-min fairness after the source that has the longest RTT receives its first BRM cell. Therefore, the above proof is conformable with the proof of single link in Theorem 1.

Using Lemma 2, 3 and 4, the FRACA algorithm converges according to the following theorem.

Theorem 2

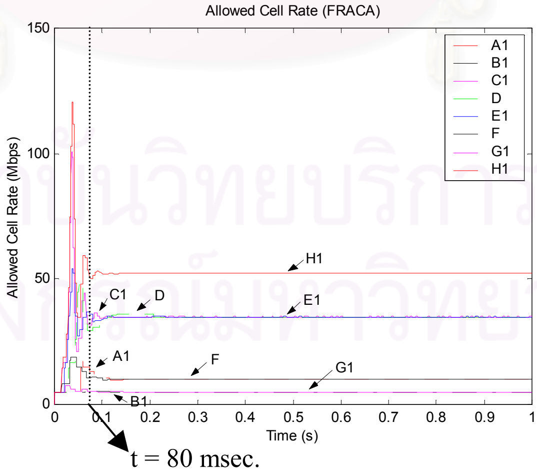
For multiple-link scenario, if all sources are persistent, the FRACA algorithm will converge to the max-min fair rate allocation in finite time.

Proof

From lemma 3 and 4, it is obvious that for multiple-link scenario with persistent sources the FRACA algorithm takes at least $\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a$ to converge to the max-min fair rate allocation in downstream bottleneck case (lemma 3) and takes a finite time, i.e. $n < \log_{1+\varepsilon} \frac{\Re_i}{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} ER_i^k \left(\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a \right)}$ averaging intervals, to achieve the max-min fair rate allocation in upstream bottleneck case (lemma 4). Hence, the FRACA algorithm will converge to max-min fair rate allocation in finite time. ■

3.4.2.1 Conformity Analysis

This section presents the conformity analysis between the analytical proof and the simulation result, in a multiple-link scenario. Let us use the simulation result of the GFC-2 configuration in Figure 3.25 a) to evaluate the consistence. For clarification, let see the result shown in Figure 3.33 (the same as Figure 3.25).



t = 80 msec.

Figure 3.33: ACRs for GFC-2 configuration

From the configuration in Figure 3.17, the longest round trip time is belong to the source B(1) that traverses all switches. The longest round trip time is 16×2.5

msec. = 40 msec. Thus, the minimum time that the algorithm will converge to the max-min fairness is 40 msec. However, the configuration also consists of the upstream bottleneck scenario. Consequently, the convergence time will be greater than the minimum time. From the result in Figure 3.33, all allocated rates converge to the max-min fairness approximately after t = 80 msec. This implies that the algorithm takes approximately 40 msec. to converge to the max-min fairness in upstream configuration, after the receiving the BRM cell of the longest RTT source. Thus, the simulation result in the case of multiple-link scenario with persistent sources confirms the analytical proof given in Section 3.4.2.

3.5 Summary

This chapter evaluates the performance of the proposed FRACA algorithm compared with the ERICA+ and E-FMMRA algorithm in various network scenarios. The FRACA is obvious to have fast transient response according to the max-min fairness, tolerance to network conditions (i.e. aggregate traffic characteristic, number of traversing ABR sessions, variation of ABR capacity and propagation delay), maximum resource utilization with controllable switch queue length. Moreover, the algorithm is fully compatible with ATM Forum Traffic Management [1] standard. In addition to simulation results, the algorithm is analytical proven to be convergent in the case that all sources are persistent. However, more cares are needed when further applying FRACA, as well as ERICA+ and FRACA, to a large-scale network due to the per-connection accounting requirement. Moreover, FRACA is not recommended to implement with a switch that has too small buffer size due to its relatively long averaging interval.

CHAPTER 4
THE NON-ZERO MCR GUARANTEE FAST RATE ALLOCATION CONGESTION AVOIDANCE (FRACA) ALGORITHM

In the last chapter, the FRACA algorithm is introduced to solve the limitations of the existing zero-MCR (Minimum Cell Rate) guarantee ABR congestion control schemes. To get a guarantee for the minimum amount of service that an user can specify a MCR, ABR service guarantees that the ACR is never less than the negotiated MCR. When MCR is zero for all sources, the available bandwidth can be allocated equally among the competing sources. This allocation achieves max-min fairness. In this chapter we treat the case of MCR that are not zero. When MCRs are non-zero, ATM Forum specification [1] recommends other definitions of fairness that allocate the excess bandwidth. The excess bandwidth is the available ABR capacity less the sum of MCRs. It may be shared equally among all competing sources or proportional to MCRs. Basically, allocations of the excess bandwidth that is proportional to the negotiated MCR required per-VC computation. Thus, it increases the complexity burden on the switches. Consequently, in this chapter the proposed FRACA algorithm is extended to support MCR plus equal-share strategy [1]. This chapter is organized as follows: Section 4.1, the existing ABR switch congestion avoidance algorithms are being evaluated. For each algorithm, the key technique is given. In addition, we will identify the key contributions and present its drawbacks. The non-zero MCR guarantee FRACA algorithm is introduced in Section 4.2. Section 4.3 presents the performance results of the proposed algorithm compared with the General Weighted Fairness (GWF) ERICA+. The non-zero MCR guarantee FRACA convergence analysis is given in Section 4.4.

4.1 Existing non-zero MCR guarantee explicit rate ABR congestion control algorithms

Although, many explicit rate ABR congestion control algorithms have been proposed during a few years [2, 3, 4, 12, 22]. Most of them are designed for zero MCR guarantees. From our survey, there are only two non-zero MCR guarantee explicit rate allocation proposed, the Generalized Max-Min (GMM) rate allocation and the General Weighted Fairness (GWF) ERICA+.

4.1.1 Generalized Max-Min (GMM) rate allocation

4.1.1.1 Key techniques

Y. T. Hou et al. [16, 17] proposed Generalized Max-Min (GMM) rate allocation for non-zero MCR ABR service. The main concept of GMM rate allocation is maximizing the minimum rate among all sessions while satisfying each session's minimum rate requirement and peak rate constraint. The iterative steps of GMM rate allocation are as follows:

1. Start the rate allocation of each session with its MCR.
2. Increase the rate of each session at a rate proportional to its MCR until either some link becomes saturated or some session reaches its PCR (Peak Cell Rate), whichever comes first.
3. For the sessions that either traverse saturated links or have reached their PCRs, remove such sessions and their associated bandwidth from the network.
4. If there is no session left, the algorithm terminates; otherwise, go back to step 2 for the remaining sessions and remaining network capacity.

Basic GMM requires global system parameters for rate allocation that is difficult to handle in the real networks. Then, Y. T. Hou et al. [16] also designed distributed-GMM rate allocation based on the Intelligent marking technique. However intelligent marking could exhibit severe oscillations and may not converge to the desired fair rate [12]. Considerable unfairness could persist when the estimation of MACR is not close to the fairshare to which the CCR values of the connections are supposed to converge. This situation could happen when some connections are bottlenecked elsewhere. In addition, Y. H. Long et al. [18] proposed the explicit rate algorithm of GMM ABR service. The algorithm aims to distinguish set of sessions bottlenecked (locally) to their MCRs, denoted as *M*, and set of sessions bottlenecked elsewhere, named as *R*, from all traversing sessions. Hence, the allocated rate is computed by

$$ER = \frac{C - \sum_{i \in R} d_i - \sum_{i \in M} m_i}{N - \|R\| - \|M\|} \qquad (4\text{-}1)$$

where *C* is the capacity of the outgoing link, *d* is “demanded rate” and *m* is “minimum rate”.

4.1.1.2 Characteristics of Generalized Max-Min (GMM) rate allocation scheme

The contribution of the Generalized Max-Min (GMM) rate allocation is as follow:

1. The computation steps are straight forwards that can reduce the implementation complexity in the switches.

The drawbacks of the Generalized Max-Min (GMM) rate allocation are as follow:

1. The algorithm assumes that the exact transmission rate is designated in the CCR field. If the aggregate traffic is not consistent, the CCR field in the RM cell may not indicate the actual transmit rate. Unfortunately the other sessions cannot utilize the unused capacity causing the system underutilized that contradicts to the main aim of ABR service category. Hence, the algorithm no longer functions efficiently in the real working environments.

2. The length of averaging interval has not defined then in case of presence of higher priority (CBR or VBR) traffic the algorithm has to be further researched.

4.1.2 The General Weighted Fairness (GWF) ERICA+

4.1.2.1 Key techniques

Vandalore et al. [19] extended the conventional ERICA+ for zero MCR requirement proposed in [4, 23], and named as GWF ERICA+. The GWF ERICA+ allocated summation of *ExcessFairshare*, which is the amount of bandwidth allocated over the MCR in a fair manner, and per connection’s MCR. In addition, *Excess Activity Level* (EAL) is employed in the modified algorithm. EAL indicates how much of the *ExcessFairshare* is actually being used by the connection. EAL is set at zero if source rate, which is indicated by Current Cell Rate field in a FRM cell, of the considered connection is less than its MCR. EAL attains the value of one when the *ExcessFairshare* is used by the connection. Actually, the reason for using EAL to indicate how much the allocated bandwidth is used by the connection is that the algorithm uses CCR field, which does not reflect the actual transmitted rate, in the rate allocation processes. Therefore in some scenarios, such as a connection is bottlenecked elsewhere or bottlenecked source, the allocated rate without EAL does not converge to the general weighted fairness.

The key steps in GWF ERICA+ are as follows:

```
***At the end of Averaging Interval:***
Total ABR Capacity ← Link Capacity – VBR Capacity - $\sum_{i=0}^{n} \min(\text{SourceRate}(i), \mu_i)$
Target ABR Capacity ← Fraction × Total ABR Capacity
Input Rate ← ABR Input Rate - $\sum_{i=0}^{n} \min(\text{SourceRate}(i), \mu_i)$
Z ← $\frac{\text{Input Rate}}{\text{Target ABR Capacity}}$

***For each VC i***
EAL(i) ← min (1, $\frac{\max(0, \text{SourceRate}(i) - \mu_i)}{\text{ExcessFairShare}(i)}$)
*SumOfWts* ← *SumOfWts* + $w_i$EAL(i)
***End for***

***For Each VC i***
*ExcessFairShare*(*i*) ← $\frac{(\text{Target ABR Capacity})w_i}{\text{SumOfWts}}$
***End for***

If (implement per-VC measure source rate option)
***For each VC i***
CCR[VC] ← $\frac{\text{Number of cell [VC]}}{\text{Averaging Interval}}$
***End for***
```

```
When a BRM is received:
VCShare ← $\frac{\max(0, \text{SourceRate}(i) - \mu_i)}{Z}$
ER ← $\mu_i$ + max (ExcessFairShare(i), VCShare)

$ER_{RM\_Cell}$ ← min ($ER_{RM\_Cell}$, ER, Target ABR Capacity)

When a FRM is received:
If (not implement per-VC measure source rate option)
        CCR[VC] ← CCR in RM Cell

When a ABR data cell is received:
If (implement per-VC measure source rate option)
        Number of cell[VC] ← Number of cell[VC] + 1
Else    Number of cell ← Number of cell + 1
```

It is obvious that the computation steps in the GWF ERICA+ [19] are no longer similar to the conventional ERICA+ [4, 23]. Consequently, the arising problems are completely different from the problems in conventional ERICA+ given in Chapter 2.

4.1.2.2 Characteristics of GWF ERICA+

The contribution of the GWF ERICA+ is as follow:

1. The algorithm is firstly support all non-zero MCR guarantee rate allocation strategies defined in standard body of ATM Forum specification [1].

The drawbacks of the GWF ERICA+ are as follow:

1. The computation steps are no longer similar to the conventional ERICA+ algorithm.

2. Although the GWF ERICA+ works well in some scenarios and parameter settings illustrated in [19], the algorithm requires per-VC rate measurement in some

scenarios (especially in bottlenecked source configuration). The per-VC computation increases the complexity burden on the switches.

3. If each session Initial Cell Rate (ICR) is low (compared with the max-min fair rate), i.e. lower than or equal to the negotiated MCR, EAL becomes zero. This causes a small value of term *SumOfWts* (probably zero). In the case, the *ExcessFairShare* and the allocated rates to all sessions become the peak cell rate leading to the severe cell loss problem at the bottlenecked switch (the aggregate ABR traffic is *n*-time larger than the available bandwidth, where *n* is the number of traversing ABR sessions).

4. The exponential averaging of *ExcessFairShare* term causes very slow transient response in the networks that the leftover bandwidth from the higher priority traffic, i.e. VBR, continually varies all the time. In the situation, the leftover bandwidth immediately changes despite that the *ExcessFairShare* term could not be immediately computed exactly. The slow transient response causes high buffer occupancy in the bottlenecked switch that may lead to the cell loss problem.

4.2 The Fast Rate Allocation Congestion Avoidance (FRACA) algorithm for non-zero MCR Guarantee

The conventional FRACA (for zero MCR requirements) in the last chapter is straightforwardly and simply modified to guarantee non-zero MCR requirement. As the results, the standout advantages over the ERICA+ and E-FMMRA is still existing. This approach is different from the GWF ERICA+, which the computation steps are no longer similar to the conventional ERICA+.

The pseudo-code for the non-zero MCR guaranteed FRACA algorithm is as follows:

***Parameters*:**

C = Link capacity

$\Re'$ = Non-ABR (higher priority) capacity

$\Re$ = ABR capacity

$\Re_n$ = Total non-bottlenecked capacity

$\beta_{s(i)}$ = Bottlenecked state of VC i

$\beta_{c(i)}$ = Non-bottlenecked capacity of VC i

ER_{max} = Maximum value of allocated rates

N_b = Number of bottlenecked sessions

α = Averaging factor

$RM_i \rightarrow ER$ = ER field in RM cell of VC i

MCR_i = MCR of VC i

SumMCR = Summation of MCR for all VCs

AR = Advertised Rate

When a BRM cell of VC i is received:

```
If ( $\beta_{s(i)}$ = bottleneck) then
        $RM_i \rightarrow ER = \min(RM_i \rightarrow ER - MCR_i, \max(AR, \frac{ER_{max}}{\rho}))$
Else    $RM_i \rightarrow ER = \min(RM_i \rightarrow ER - MCR_i, AR)$
$ER_{max} = (1.0 - \alpha) \times ER_{max} + \alpha \times \max(RM_i \rightarrow ER, \frac{ER_{max}}{\rho})$
If ($AR \le RM_i \rightarrow ER$) then
        $N_b = N_b - \beta_{s(i)} + 1$
        $\beta_{s(i)} = 1$
        $RM_i \rightarrow ER = RM_i \rightarrow ER$ + adjusted ER
        $\Re_n = \Re_n - \beta_{c(i)}$
        $\beta_{c(i)} = 0$
Else    $N_b = N_b - \beta_{s(i)}$
        $\beta_{s(i)} = 0$
        $\Re_n = \Re_n - \beta_{c(i)} + RM_i \rightarrow ER$
        $\beta_{c(i)} = RM_i \rightarrow ER$
If ($N_b > 0$) then
        $AR = \frac{\Re - \Re_n}{N_b}$
Else    $AR = AR + (\Re - \Re_n)$
```

```
        $RM_i \rightarrow ER = RM_i \rightarrow ER + MCR_i$

***When average interval is expired:***
        ***For Each VC i***
            $SumMCR = SumMCR + MCR_i$
        ***End for***
        $\Re = C - \Re' - SumMCR$
        $\rho = \frac{\text{ABR Input Rate}}{\Re \times \text{Qfactor}}$
        $\text{adjusted ER} = \frac{(\text{Qfactor} - 1) \times \Re}{N_b}$

***When a ABR data cell is received:***
Number of cell = Number of cell + 1
```

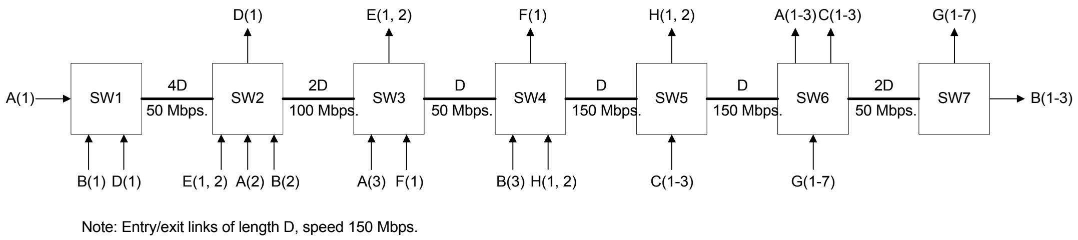
For queue control feature, the same Qfactor value as proposed in Table 3.1 is employed in the rate allocation process.

4.3 Performance evaluation of non-zero MCR guarantee explicit rate ABR congestion avoidance algorithm

This section examines the efficiency, fairness, transient and steady state performance and finally its adaptation to variable capacity and various source traffic models in non-zero MCR guarantee environments.

4.3.1 GFC-2 with MCR guarantee scenario

The scenario is modified from the conventional GFC-2 [8] for non-zero MCR requirement. The configuration is a combination of upstream and parking-lot scenario. The configuration and parameter setting are shown in Figure 4.1.



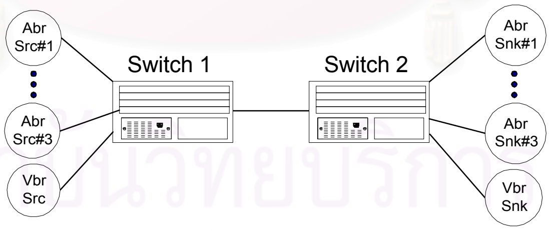
Parameter setting:

- All sources are persistent, PCR = 150.0 Mbps., ICR = 5.0 Mbps.
- MCR A(1, 2, 3) = (0, 2, 3 Mbps.), B(1, 2, 3) = (5, 1, 2 Mbps.), C(1-3) = 0.5 Mbps., D = 2 Mbps., E(1, 2) = 5 Mbps., F = 0 Mbps., G(1-7) = 0.5 Mbps and H(1, 2) = 1 Mbps.
- Propagation delay D = 2.5 msec.

Figure 4.1: The GFC-2 with MCR guarantee scenario

4.3.2 Source bottleneck plus VBR scenario

The scenario is used to illustrate the effects of a bottlenecked source, i.e. source sends data lower than the allocated rate, and continually change of the leftover bandwidth (from higher priority traffic category) to the performance of the rate allocation algorithms. The configuration and parameter setting are shown in Figure 4.2. In addition, the VBR cell rate is shown in Figure 4.3.



Parameter setting:

- Source 2 and 3 are persistent sources while source 1 sends data at 15.0 Mbps. during 0-400 msec, irrespective of the ACR, and becomes persistent source during 400-800 msec.
- All sources ICR is 5.0 Mbps.
- MCR of source 1, 2 and 3 are 5.0, 10.0 and 15.0 Mbps., respectively
- All link propagation delay = 5 msec. and have capacity = 150.0 Mbps.
- Output switch buffer size = 10,000 cells.

– VBR source is an on-off source that traffic pattern shown in Figure 4.3.

Figure 4.2: Source bottleneck plus VBR scenario

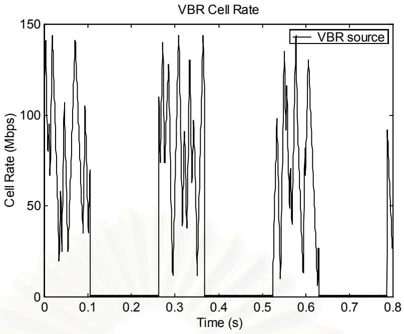
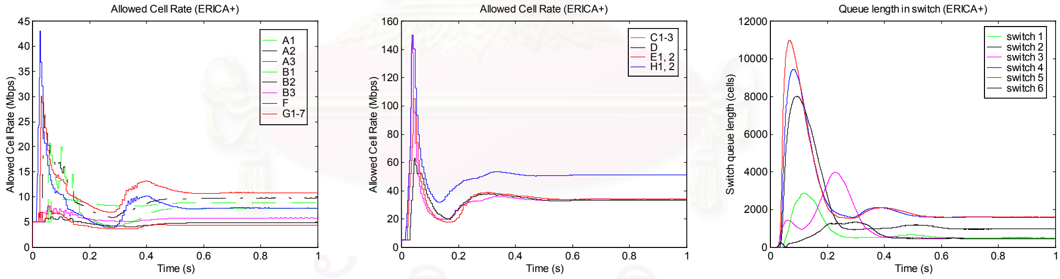


Figure 4.3: VBR cell rate

4.3.3 Performance results

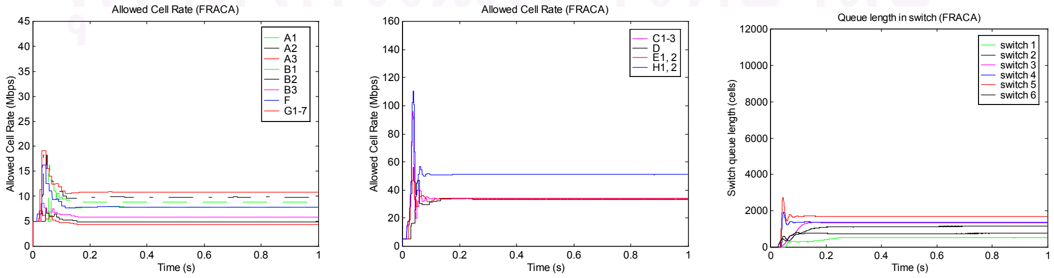
This section gives the performance results of the non-zero MCR guarantee FRACA algorithm compared with the GWF ERICA+ under the scenarios mentioned above.

Figure 4.4 and 4.5 illustrate the results of the GFC-2 with MCR guarantee for the GWF ERICA+ and FRACA, respectively.



(a) Allowed Cell Rate (b) Switch queue length

Figure 4.4: Results for the GFC-2 with MCR guarantee (GWF ERICA+)



(a) Allowed Cell Rate (b) Switch queue length

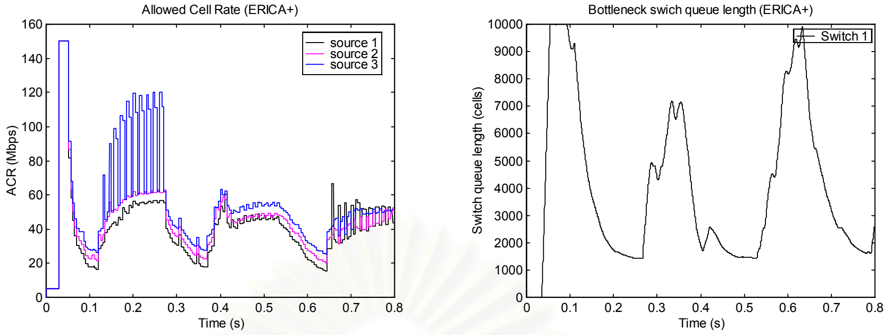
Figure 4.5: Results for the GFC-2 with MCR guarantee (FRACA)

The followings are the expected rates allocated to all sessions according to MCR plus equal share strategy.

A1, 2 and 3 VCs get	= 7.825, 9.825 and 10.825 Mbps., respectively
B1, 2, and 3 VCs get	= 8.85, 4.85 and 5.85 Mbps., respectively
C VCs each get	= 33.99 Mbps.
D VC gets	= 33.325 Mbps.
E VCs each get	= 34.325 Mbps.
F VC gets	= 7.825 Mbps.
G VCs each get	= 4.35 Mbps.
H VCs each get	= 50.9875 Mbps.

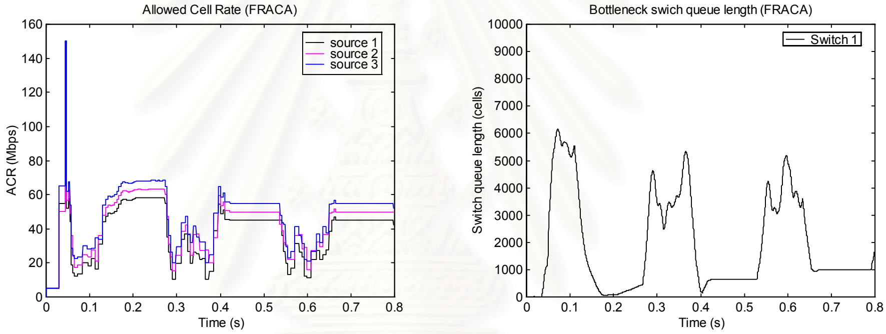
For simplicity, the ACR graphs in Figure 4.4 a) and 4.5 a) are separated into two groups, i.e. the lower rate grouping (VCs A, B, F and G) and the higher rate grouping (VCs C, D, E and H). It is obvious that both algorithms converge to the max-min rates despite that the convergence time of the GWF ERICA+ is longer than that of the FRACA, approximate 500 msec compared with 100 msec. The main reason is that the allocated rates for all sessions by GWF ERICA+ rapidly grow during transient period due to the problem from the low initial cell rate (5.0 Mbps. compared to the expected rates) as the reason cited in Section 4.1.2.2. Consequently, the over allocated rates cause the high buffer occupancy in the bottlenecked switches (as shown in Figure 4.4 b). For FRACA algorithm, it is obvious from Figure 4.5 that the allocated rates rapidly converge to the max-min rates with constant controllable queue level. The switch queue behavior implies that the allocated rates are the max-min fair rates.

The results of the source bottleneck + VBR scenario for the GFW ERICA+ and FRACA are shown in Figure 4.6 and 4.7, respectively.



(a) Allowed cell rate (b) Switch queue length

Figure 4.6: Results for source bottleneck plus VBR (GWF ERICA+)



(a) Allowed cell rate (b) Switch queue length

Figure 4.7: Results for source bottleneck plus VBR (FRACA)

The followings are the expected rates allocated to all sessions (during off period of VBR traffic) according to MCR plus equal share strategy.

Source 1 gets = 60.0 and 45.0 Mbps. during 0-400 msec. and 400-800 msec., respectively

Source 2 gets = 65.0 and 50.0 Mbps. during 0-400 msec. and 400-800 msec., respectively

Source 3 gets = 70.0 and 55.0 Mbps. during 0-400 msec. and 400-800 msec., respectively

For a robustness point of view, the performance of a congestion avoidance algorithm should not be affected by continually changing in the leftover capacity from higher priority traffic. In addition, the algorithm should tolerate to the CCR error in source bottleneck scenario, i.e. source sends data below the allocated rate, which the CCR field does not reflect the actual transmitted rate. It is obvious from the results in Figure 4.6 that the allocated rates for all sessions by GWF ERICA+ rapidly grow during transient period due to the problem from the low initial cell rate, 5.0 Mbps. compared to the expected rates. Consequently, this causes the high buffer occupancy in the bottlenecked switches. Moreover, the allocated rates by the GWF ERICA+ diverge from the max-min rates (during 0-400 msec.) due to the CCR error. The rate oscillation disappears after 400 msec. despite that the allocated rates slowly converge to the max-min rates. The main reason for slow response is that the algorithm employs the exponential average of the *ExcessFairshare* term [19]. Hence, if the leftover bandwidth immediately increases despite that the *ExcessFairshare* could not be immediately computed exactly. Consequently, the slow response (especially during on period of VBR traffic) causes high buffer occupancy. The FRACA shows the better performance, compared with the GWF ERICA+. Result in Figure 4.7 shows that the algorithm has a fast transient response and all the allocated rates are free from oscillation. In addition, obviously the algorithm immediately reallocates the correct rates to all sessions after the overload detection due to the aggregate traffic of source 1 at t=400 msec. Consequently, the switch queue level is kept under control during steady state period.

4.4 The non-zero MCR guarantee FRACA convergence analysis

This section presents a mathematical proof of the convergence of the non-zero MCR guarantee FRACA algorithm. It is assumed that the numbers of sessions traversing links do not change during the convergence period. The proof is straightforwardly modified from the convergence analysis of the zero MCR guarantee FRACA algorithm given in Section 3.4. Unless otherwise noted, the definitions and notations are defined following Section 3.4.

Additional definitions and notations

– $\Re_i$ is the capacity of link i.

– ζ_i is the left over capacity from MCR of all sessions which traversing link i, i.e. $\zeta_i = \Re_i - \sum_{i \in (\Im_i \cup \overline{\Im_i})} MCR_i$.

Following the proof in Section 3.4 and using the term ζ_i replace $\Re_i$, the max-min fair rates for the sessions in level-j bottlenecked grouping (j ≥ 1) becomes

$$r_j = \min_{i \neq \ell_{b_1}, \ell_{b_2}, \ldots, \ell_{b_{j-1}}} \left\{ \frac{\zeta_i - \sum_{m=1}^{j-1} \sum_{k \in (\beta_m \cap (\Im_i \cup \overline{\Im_i}))} (r_k + MCR_k)}{N_i - \sum_{m=1}^{j-1} n(\beta_m \cap (\Im_i \cup \overline{\Im_i}))} \right\} \tag{4-2}$$

The non-zero MCR guarantee FRACA algorithm is proved to be convergent according to the followings subsequent convergent arguments.

Lemma 1

The steady state fairshare for bottlenecked sessions at link i is greater than or equal to the summation of steady state max-min equal-share rate and its corresponding MCR.

$$ER_i^k \geq r_{b_i} + MCR_k \qquad \cdots \ \forall k \in \Im_i \tag{4-3}$$

Proof

The max-min equal-share rate (exclude corresponding MCR) for bottlenecked sessions at link i, i.e. level-b$_i$ bottlenecked grouping, is computed as

$$r_{b_i} = \frac{\zeta_i - C_{\overline{\Im_i}}}{\lambda_i} \tag{4-4}$$

If all lower level bottlenecked at link i have stabilized at their max-min rate, these sessions are non-bottlenecked at the link, by definition, that is $[\beta_k \cap (\Im_i \cup \overline{\Im_i})] \subseteq \overline{\Im_i}$ for $k < b_i$. The rest sessions should receive r_{b_i} as feedback equal-share rate. However, some sessions may be bottlenecked by their PCR, i.e. $(r_{b_i - 1} + MCR_{k'}) < ER_i^{k'} < (r_{b_i} + MCR_{k'})$ for $k' \in (\overline{\Im_i} \cap \beta_{b_i})$. Hence, the max-min equal-share rate to the other sessions is computed as

$$\begin{aligned} ER_i^k &= \frac{\zeta_i - \sum_{j=1}^{b_i-1} C_{\overline{\Im_i^j}} - \sum_{k' \in \left(\overline{\Im_i} \cap \beta_{b_i}\right)} ER_i^{k'}}{N_i - \sum_{j=1}^{b_i-1} \overline{\lambda_i^j} - n\left(\beta_{b_i} \cap \left(\Im_i \cup \overline{\Im_i}\right)\right)} \\ &= \frac{\zeta_i' - \sum_{k' \in \left(\overline{\Im_i} \cap \beta_{b_i}\right)} ER_i^{k'}}{N_i' - n\left(\beta_{b_i} \cap \left(\Im_i \cup \overline{\Im_i}\right)\right)} \\ &= \frac{\zeta_i' - C_{\overline{\Im_i'}}}{N_i' - \overline{\lambda_i'}} = \frac{\zeta_i' - C_{\overline{\Im_i'}}}{\lambda_i'} \end{aligned} \tag{4-5}$$

where $\zeta_i' = \zeta_i - \sum_{j=1}^{b_i-1} C_{\overline{\Im_i^j}}$ is the left over bandwidth from level-1 through level-$b_i - 1$non-bottlenecked sessions traversing link i.

$N_i' = N_i - \sum_{j=1}^{b_i-1} \overline{\lambda_i^j}$ is the number of sessions left over from level-1 through level-$b_i - 1$ non-bottlenecked sessions traversing link i.

$C_{\overline{\Im_i'}} = \sum_{k' \in \left(\overline{\Im_i} \cap \beta_{b_i}\right)} ER_i^{k'}$ is the total bandwidth usage by level-b_i non-bottlenecked sessions traversing link i that are bottlenecked by their PCRs.

$\overline{\lambda_i'} = n\left(\beta_{b_i} \cap \left(\Im_i \cup \overline{\Im_i}\right)\right)$ is the number of level-b_i non-bottlenecked sessions traversing link i that are bottlenecked by their PCRs.

The link i becomes bottlenecked with respect to these sessions. From definition and iterative computation, the computed $ER_i^k (\forall k \in \Im_i)$ is always higher than or equal to, i.e. if $C_{\overline{\Im_i'}} = \overline{\lambda_i'} = 0$, the allocated rate of level-$b_i$bottlenecked sessions traversing link i. ■

Lemma 2

For multiple-link scenario, if all sources are persistent, the minimum time that the FRACA algorithm will converge according to the max-min fairness is $\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a$.

Proof

This is the case of downstream bottleneck scenario. During backward path of the first BRM cell, the expected non-bottlenecked sessions are sequentially marked as the lower-level bottlenecked grouping at the less congested link. Then the intermediate switches compute and allocate the advertised max-min equal-share rate

(4-5) to all bottlenecked sessions at the end of the averaging interval after the longest RTT. ■

Lemma 3

For multiple-link scenario, if all sources are persistent, the FRACA algorithm will converge according to the max-min fairness within $n < \log_{1+\varepsilon} \dfrac{\Re_i}{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} \left(ER_i^k \left(\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a \right) + MCR_k \right)}$ averaging intervals.

Proof

This is the case of upstream bottleneck scenario. The important task is the way to let the downstream switches know the bottleneck condition in the upstream switches because the algorithm works only in the backward direction. The load factor, defined as the ratio of ABR input rate to ABR Capacity (the queue control factor is relaxed), is introduced to solve the problem. Let t' denote the time that the FRACA algorithm converges according to MCR plus equal share rate allocation, i.e. $ER_i^k(t') = ER_i^k$ according to (4-5), and let ρ_{i_t} be the load factor of link i during time interval $(t \to t + t_a)$.

Basically in the case of upstream bottleneck, initially, the AR at the downstream link cannot fully utilize the link capacity. The AR will be continually increased until the downstream link is fully utilized. That is

$$1 = \rho_{i_{t'}} > \rho_{i_{t'-t_a}} > \rho_{i_{t'-2t_a}} > \ldots > \rho_{i_{\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a}} \tag{4-6}$$

The system will converge to the max-min rate allocation if ρ_{i_t} becomes one $(R_i = \Re_i)$. In other words, the equal-share rates for all bottlenecked sessions are the max-min equal-share rates (4-5). Equation (4-6) could be rewritten as

$$1 = \rho_{i_{t'}} = \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} \left(ER_i^k(t') + MCR_k \right)}{\Re_i} > \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} \left(ER_i^k(t' - t_a) + MCR_k \right) \Big/ \rho_{i_{t'-t_a}}}{\Re_i} > \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} \left(ER_i^k(t' - 2t_a) + MCR_k \right) \Big/ \rho_{i_{t'-t_a}} \cdot \rho_{i_{t'-2t_a}}}{\Re_i} > \ldots$$
$$> \frac{\sum_{j=1}^{b_i} \sum_{k \in (\Im_i \cap \beta_j)} \left(ER_i^k \left(\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a \right) + MCR_k \right) \Big/ \rho_{i_{t'-t_a}} \cdot \rho_{i_{t'-2t_a}} \cdot \ldots \cdot \rho_{i_{\left\lceil \frac{t_{\max}}{t_a} \right\rceil \cdot t_a}}}{\Re_i} \tag{4-7}$$

It is obvious that the load factor $\rho_{t'-t_a}$ is less than $\frac{1}{1+\varepsilon}$ for small ε. Hence,

$$\rho_{i_{t'}} > \frac{\sum_{j=1}^{b_i}\sum_{k\in(\Im_i\cap\beta_j)}\left(ER_i^k\left(\left\lceil\frac{t_{\max}}{t_a}\right\rceil\cdot t_a\right)+MCR_k\right)\Big/\rho_{i_{t'-t_a}}\cdot\rho_{i_{t'-2t_a}}\cdot\ldots\cdot\rho_{i_{\left\lceil\frac{t_{\max}}{t_a}\right\rceil\cdot t_a}}}{\Re_i} > \frac{\sum_{j=1}^{b_i}\sum_{k\in(\Im_i\cap\beta_j)}\left(ER_i^k\left(\left\lceil\frac{t_{\max}}{t_a}\right\rceil\cdot t_a\right)+MCR_k\right)\Big/\left(\frac{1}{1+\varepsilon}\right)^n}{\Re_i} \tag{4-8}$$

where *n* is the number of averaging intervals (cycles) that the FRACA algorithm converges according to the max-min fair rate after the session, which has the longest RTT, receives the first BRM cell, i.e. $n=\left\lceil\frac{t'-t_{\max}}{t_a}\right\rceil$. From (4-7) and (4-8), it is obvious that the FRACA algorithm will converge according to the following constraint

$$\frac{\sum_{j=1}^{b_i}\sum_{k\in(\Im_i\cap\beta_j)}\left(ER_i^k\left(\left\lceil\frac{t_{\max}}{t_a}\right\rceil\cdot t_a\right)+MCR_k\right)}{\left(\frac{1}{1+\varepsilon}\right)^n} < \Re_i$$

Therefore,

$$n < \log_{1+\varepsilon}\frac{\Re_i}{\sum_{j=1}^{b_i}\sum_{k\in(\Im_i\cap\beta_j)}\left(ER_i^k\left(\left\lceil\frac{t_{\max}}{t_a}\right\rceil\cdot t_a\right)+MCR_k\right)} \tag{4-9}$$

The above equation implies that the FRACA algorithm will converge to the max-min fair rate in finite time. ■

Using Lemma 1, 2 and 3, the non-zero MCR guarantee FRACA algorithm converges according to the following theorem.

Theorem 1

For a scenario that all sources are persistent, the FRACA algorithm will converge to max-min fair rate allocation in finite time.

Proof

From lemma 2 and 3, it is obvious that the algorithm takes $\left\lceil\frac{t_{\max}}{t_a}\right\rceil\cdot t_a$ to converge to the max-min fair rate allocation in downstream bottleneck scenario (lemma 2) and takes a finite time, i.e. $n<\log_{1+\varepsilon}\frac{\Re_i}{\sum_{j=1}^{b_i}\sum_{k\in(\Im_i\cap\beta_j)}\left(ER_i^k\left(\left\lceil\frac{t_{\max}}{t_a}\right\rceil\cdot t_a\right)+MCR_k\right)}$, to achieve the max-min fair rate allocation in upstream bottleneck scenario (lemma 3). Hence, the FRACA algorithm will converge to max-min fair rate allocation in finite time. ■

4.5 Summary

This chapter investigates the problems arising in the existing non-zero MCR guarantee congestion avoidance algorithms for point-to-point ABR service in ATM networks. For robustness issue, it is shown that the GWF ERICA+ algorithm no longer functions efficiently in the specified environments. The non-zero MCR guarantee FRACA algorithm is proposed for handling MCR plus equally share rate allocation strategy. The algorithm is obvious to have very fast transient response according to the max-min fairness, tolerance to network conditions (i.e. aggregate traffic characteristics and leftover bandwidth), maximum resource utilization with controllable switch queue length. In addition, the algorithm is fully compatible with the ATM traffic management [1] standard.

CHAPTER 5
THE EFFECTS OF CONGESTION AVOIDANCE ALGORITHM ON THE PERFORMANCE OF POINT-TO-MULTIPOINT ABR SERVICE

Point-to-multipoint is a new challenging strategy for emerging applications in ATM networks. It is obvious that most of today data applications perform point-to-multipoint behavior i.e. Internet, data transfer server. Using simulation results, this chapter will illustrate that the performance of point-to-multipoint ABR connections is also depend on a congestion avoidance algorithm that is formerly designed for point-to-point scenarios (instead of the consolidation algorithms). This chapter is organized as follows. Section 5.1 states the key concepts and problems arising in point-to-multipoint ABR connections. Section 5.2 gives the reasons how the proposed FRACA algorithm can improve the performance of point-to-multipoint ABR connections. Section 5.3 presents the performance results of the FRACA algorithm in point-to-multipoint scenarios compared with the ERICA+ both in zero and non-zero MCR guarantee environments.

5.1 Point-to-multipoint consolidation algorithms in ABR service

Generally, problems in point-to-multipoint scenarios are more complicated than point-to-point scenarios. For point-to-multipoint connections, switches at the branch points consolidate the information in backward RM cells that are received from downstream switches or destinations and forward the consolidated information to their upstream switches or sources. The model for point-to-multipoint ABR connections is shown in Figure 5.1.

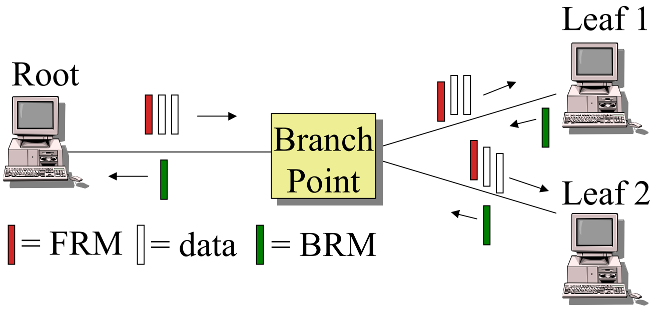


Figure 5.1: Model of point-to-multipoint ABR connections

Typically, the goals for designing consolidation algorithm are accuracy of aggregated feedback information while maintaining fast transient response. The algorithms that attempt to provide fast response will suffer from severe rate oscillation due to consolidation noise. On the other hand, algorithms that attempt to maximize accuracy of feedback information will tend to be slow in providing feedback. A few modern consolidation algorithms could achieve the designed goals but the implementation complexity is very high. In addition, they introduce threshold setting problem, i.e. the conditions to immediately pass back the BRM without waiting for all branch BRM cells. Unfortunately all designed consolidation algorithms [6, 20, 28, 29, 30] are examined under the same explicit rate switch algorithm, ERICA without queue control option. This dissertation pioneers the study of the effects of an explicit rate switch algorithm to point-to-multipoint connections based on the FRACA as a sample switch algorithm.

5.1.1 Not wait for all (Ren) consolidation algorithm

Ren et al. [31] proposed a simple consolidation algorithm (algorithm 3 in [20]). The main idea is that the BRM cell, that is received from a branch, is immediately passed back to the source after a FRM has been received. The feedback explicit rate in the consolidated BRM cell is the minimum explicit rate reported in the BRM cells received from the downstream branches since the last BRM cell was sent.

5.1.1.1 Key techniques

The key steps of the consolidation algorithm are as follows:

```
Upon receiving a FRM cell:
1. Multicast FRM cell to all participating branches
2. Let AtleastOneFRM = 1
Upon receiving a BRM cell:
1. Let MER = min(MER, ER from BRM cell)
       MCI = MCR OR CI from BRM cell
       MNI = MNI OR NI from BRM cell
2. If AtLeastOneFRM then
       A. Pass the BRM with ER = MER, CI = MCI, NI = MNI to the source
       B. Let MER = PCR, MCI = 0, MNI = 0
       C. Let AtLeastOneFRM = 0
```

Else Discard the BRM cell

When a BRM is to be scheduled

Let ER = min(ER, ER calculated by rate allocation algorithm for all branches)

5.1.1.2 Characteristics of Not wait for all (Ren) consolidation algorithm

The contributions of the not wait for all consolidation algorithm are as follows:

1. The computation complexity is low and straightforward.
2. The transient response is very fast.

The drawback of the not wait for all consolidation algorithm is as follows:

1. The algorithm experiences high consolidation noise due to the lack of the information from all branches.

The algorithm was modified for working with explicit rate switch algorithm, ERICA, and named algorithm 3 in [20]. When a consolidated BRM cell is to be scheduled, the feedback explicit rate is chosen as the minimum rate between ER in BRM cell and the allocated rate for all branches by the switch algorithm. However, a single bit (congestion indication - CI) notification is still employed for queue control objective. Simulations results in [20] indicate that the consolidation algorithm exhibits fast transient response but the algorithm experiences heavy consolidation noise due to inaccurate feedback leading to serious oscillation of allowed cell rate.

5.1.2 Immediate rate calculation (Fahmy) consolidation algorithm

Fahmy et al. introduced several consolidation algorithms. This section focuses on “Immediate rate calculation” or algorithm 7 in [20]. This algorithm shows the best performance, i.e. fast transient response while eliminating the consolidation noise problem among all algorithms. Anyway the improved performance should be traded off with higher implementation complexity. The key idea of the algorithm is that the employed explicit rate switch algorithm works whenever a BRM cell is received, not just when a consolidated BRM cell is being sent. This technique allows a switch at the branch point immediately detect overload condition at the branch point. If the calculated ER, when a BRM cell is received, is lower than multiplication between the set threshold and the last allocated ER, the BRM cell is passed to the corresponding upstream switch or source. Otherwise, a BRM cell is sent back after receiving BRM cells from all branches.

5.1.2.1 Key techniques

The key steps of the consolidation algorithm are as follows:

```
Upon receiving a FRM cell:
1. Multicast FRM cell to all participating branches
2. Let FRMminusBRM = FRMminusBRM + 1
Upon receiving a BRM cell from branch i;
1. Let SendBRM = 0
2. Let Reset = 1
3. If NOT $BRMReceived_i$ then
        A. Let $BRMReceived_i$ = 1
        B. Let NumberOfBRMsReceived = NumberOfBRMsReceived + 1
4. Let  MER = min(MER, ER from BRM cell)
        MCI = MCR OR CI from BRM cell
        MNI = MNI OR NI from BRM cell
5. Let MER = min(MER, minimum ER calculated by rate allocation algorithm for all
   branches)
6. If MER $\geq$ LastER AND SkipIncrease > 0 AND NumberOfBRMsReceived is equal
   to NumberOfBranches then
        A. Let SkipIncrease = SkipIncrease – 1
        B. Let NumberOfBRMsReceived = 0
        C. Let BRMReceived = 0 FOR all branches
   Else If MER < (Threshold $\times$ LastER) then
        A. If NumberOfBRMsReceived < NumberOfBranches then
            1. Let SkipIncrease = SkipIncrease + 1
            2. Let Reset = 0
        B. Let SendBRM = 1
   Else If NumberOfBRMsReceived is equal to NumberOfBranches then
          Let SendBRM = 1
7. If SendBRM then
        A. Pass the BRM with ER = MER, CI = MCI, NI = MNI to the source
        B. If Reset then
            1. Let MER = PCR, MCI = 0, MNI = 0
            2. Let NumberOfBRMsReceived = 0
            3. Let BRMReceived = 0 FOR all branches
        C. Let FRMminusBRM = FRMminusBRM – 1
  Else Discard the BRM cell
When a BRM is to be scheduled:
1. Let ER = min(ER, ER calculated by rate allocation algorithm for all branches)
2. Let LastER = ER
```

5.1.2.2 Characteristics of Immediate rate calculation (Fahmy) consolidation algorithm

The contributions of the immediate rate calculation consolidation algorithm are as follows:

1. The algorithm has fast transient response while the consolidation noise is completely eliminated during steady state.

2. The algorithm includes some features of the explicit rate allocation scheme (for point-to-point scenario) in the consolidation process.

The drawbacks of the immediate rate calculation consolidation algorithm are as follows:

1. The computation complexity is very high.

2. The optimum value of the "threshold" has to be further studied for working in the real environment that all parameters continually change all the time.

In this section, the consolidation algorithms are modified at the branch points in order to improve algorithms performance for supporting explicit rate switch algorithms which employ their own queue control function, i.e. ERICA+ and FRACA. All previously designed consolidation algorithms apply congestion indication (CI) flag and no increase (NI) flag in RM cells in order to control switches queue level similar to the early relative rate (RR) switch algorithm in point-to-point connection. The solution to solve the severe oscillation problem of switches queue length is to set target utilization of switches below 100%, i.e. 90% or 95% for WAN and LAN configuration [20], respectively. However, the solution causes system underutilized that contradicts to the main aim of ABR category. Hence, the switches queue level should be handled by queue control option of the switch algorithm rather than setting flag in consolidation algorithm.

5.2 Effects of point-to-point rate allocation algorithm to performance of point-to-multipoint ABR service

Most of the researches in point-to-multipoint ABR connections issue [6, 20, 28, 29, 30] use the basic ERICA (without queue control function) as the rate allocation algorithm. Many consolidation algorithms have been proposed based on the basic ERICA algorithm. Although the modern consolidation algorithms exhibit fast transient response while the consolidation noise is completely eliminated but the implementation complexity is very high. Consequently it is not appropriate for implementing in the switch working in the real environment. This dissertation pioneers that type of the rate allocation algorithm also affects performance of the point-to-multipoint ABR connections. We use our purposed rate allocation algorithm, FRACA, as a sample switch algorithm to evaluate the point-to-multipoint performance.

Remember that the FRACA algorithm has to copy the ER field of the received BRM cell in order to mark a connection as bottlenecked or non-bottlenecked connection. If the connection is a bottlenecked one, the new AR is recomputed. The bottlenecked connection will be received at most the AR as the feedback rate during the backward direction of a RM cell. For presentation purpose, let see the Figure 5.2 as an example scenario.

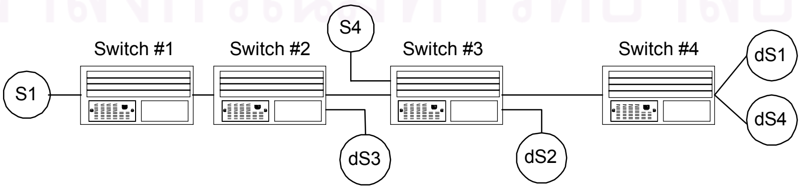


Figure 5.2: An example scenario of point-to-multipoint ABR connections

The FRACA algorithm at the output port of switch#2 copies the ER field from consolidated BRM cell to the switch's memory for marking connection purpose. Hence, the connections that are traversing link between switch#2 and 3 is marked as a non-bottlenecked connections (at switch#2) after the arrival of a BRM cell from dS4. The worst case is the situation that the consolidated BRM cell (after rate allocation of switch algorithm) is belonging to dS3 due to consolidation noise. However, the consolidated BRM cell has to be re-allocated the max-min rate by the rate allocation algorithm that computes the appropriate rates for all branches as the model shown in Figure 5.3.

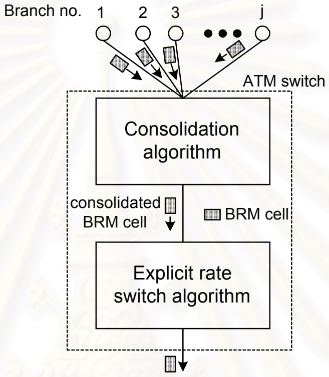


Figure 5.3: Model of consolidation and BRM cell scheduling at a branch point

Hence, the explicit rate in consolidated BRM cell is always reduced to the non-bottlenecked rate (from dS4) either to the consolidation algorithm or the rate allocation algorithm. This methodology is opposite to the process in the ERICA+ algorithm that stores and allocates rate according to the fairshare and ExcessFairshare, for the ERICA+ and GWF ERICA+, accordingly. Thus, the allocated rate by the ERICA+ algorithm at switch#2 is always the peak cell rate, i.e. 149.76 Mbps. If the consolidated BRM cell is the noise one, i.e. BRM cell from dS3, the allowed cell rate of S1 is immediately assigned to the peak cell rate value. If the consolidated BRM cell is passed from the BRM cell of dS4, the allocated rate is then reduced to the max-min rate at the bottlenecked link between switch#3 and 4, i.e. approximate 75.0 Mbps. This causes unacceptable oscillation of the allowed cell rate of S1 when running Ren consolidation algorithm over the ERICA+ in point-to-multipoint scenarios.

For eradication the consolidation noise, the consolidation algorithms that run over the ERICA+, themselves, have to eliminate the noise before passing the consolidation BRM cell to the ERICA+. As the results, the consolidation algorithms those run over ERICA+ must have a high computation complexity – compared to the consolidation algorithms that run over FRACA.

5.3 Performance evaluation of point-to-multipoint ABR connections

To evaluate the effect of the rate allocation algorithm to the performance of point-to-multipoint ABR connections, let us use 2 network scenarios to demonstrate.

5.3.1 Point-to-multipoint ABR connections with zero-MCR requirement

The first scenario is the chain scenario in [20]. In the scenario, ABR source 1 (S1) has multipoint connections to destination 1, 2 and 3 (dS1, dS2 and dS3) while source 4 (S4) establishes a point-to-point connection to destination 4 (dS4). S1 and S4 have zero MCR requirement and their Initial Cell Rate (ICR) are set equally to 149.76 Mbps. Hence, the max-min rate to source 1 and 4 is approximate 75.0 Mbps at steady state. The point-to-multipoint chain scenario is shown in Figure 5.4.

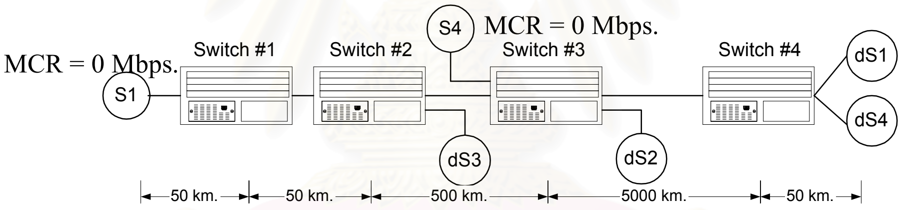


Figure 5.4: Point-to-multipoint chain scenario

5.3.2 Point-to-multipoint ABR connections with non-zero MCR guarantee

This scenario is employed to evaluate the effects of the non-zero MCR guarantee explicit rate ABR congestion avoidance algorithm on the performance of point-to-multipoint ABR connections. The configuration is similar to the point-to-multipoint chain scenario despite that the MCR requirement of source 1 and 4 is 20.0 and 30.0 Mbps., respectively. Moreover, we also test the effects of source-bottleneck on the performance of point-to-multipoint connections then source 1, bottlenecked source, always send data at 30.0 Mbps. irrespective of the ACR. The equal share of available bandwidth of the bottlenecked link (output link of switch #3) is $\frac{150.0-20.0-30.0}{2}$ = 50.0 Mbps. but the source 1 uses only 10.0 Mbps. of this available bandwidth. Hence, the bottlenecked switch should allocate the unused bandwidth (90.0 Mbps.) to all sources. Consequently, the max-min rates to source 1 and 4 are approximate 110.00 and 120.00 Mbps., respectively, according to MCR plus equal share rate allocation strategy. The scenario is shown in Figure 5.5.

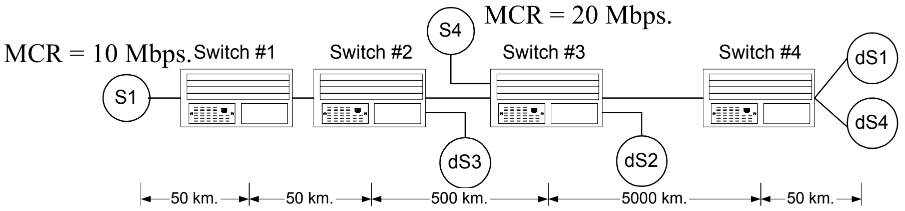
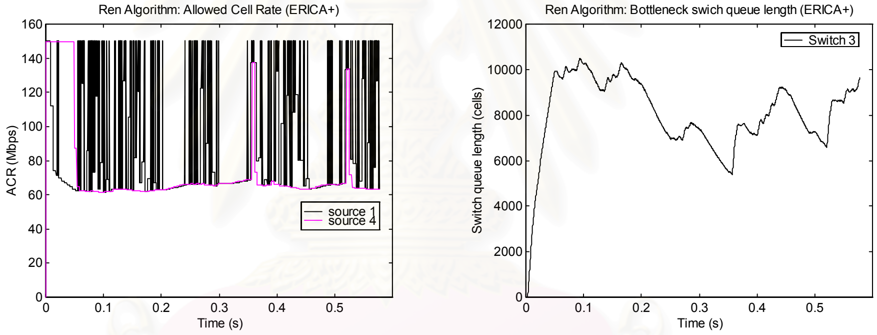


Figure 5.5: Point-to-multipoint with non-zero MCR guarantee explicit rate ABR congestion avoidance algorithm and source-bottleneck scenario

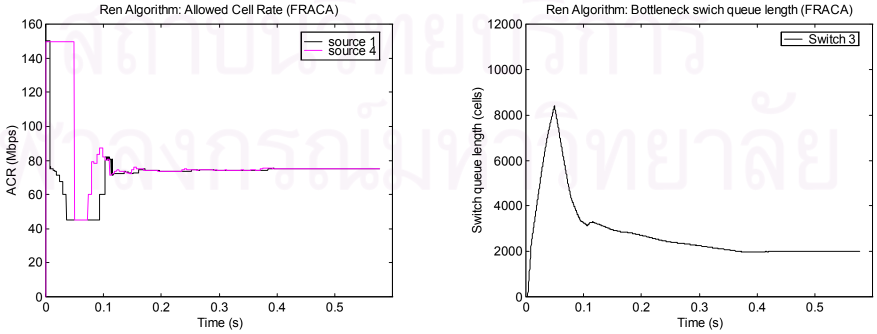
5.3.3 Performance results

This section gives the performance results of the FRACA algorithm in point-to-multipoint ABR connections under the scenarios mentioned above. The performance is also compared to the ERICA+ and GWF ERICA+.



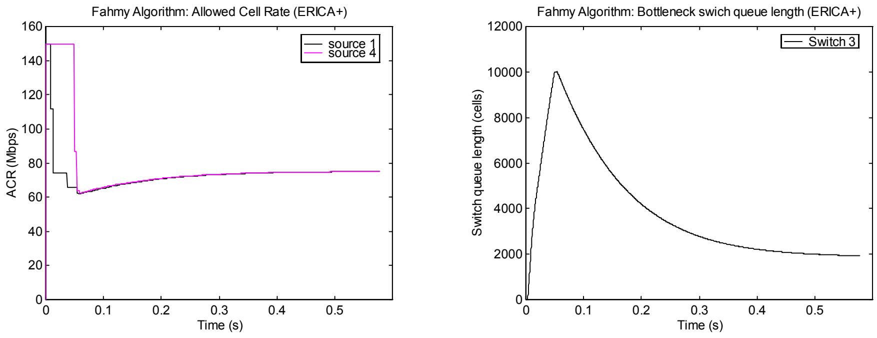
(a) Allowed cell rate (b) Switch queue length

Figure 5.6: Results for point-to-multipoint chain scenario [Ren Algorithm – ERICA+]



(a) Allowed cell rate (b) Switch queue length

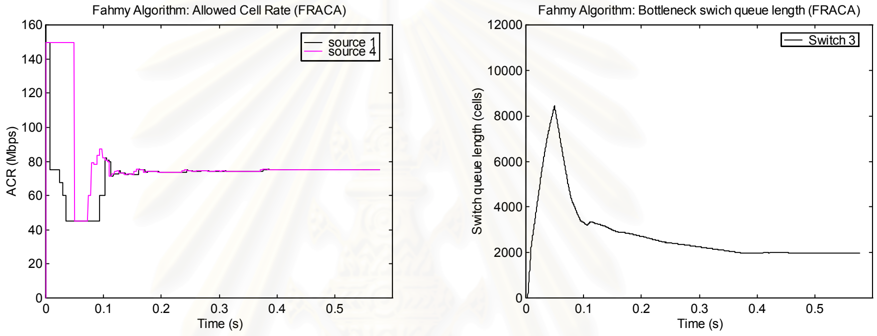
Figure 5.7: Results for point-to-multipoint chain scenario [Ren Algorithm – FRACA]



(a) Allowed cell rate (b) Switch queue length

Figure 5.8: Results for point-to-multipoint chain scenario

[Fahmy Algorithm – ERICA+]



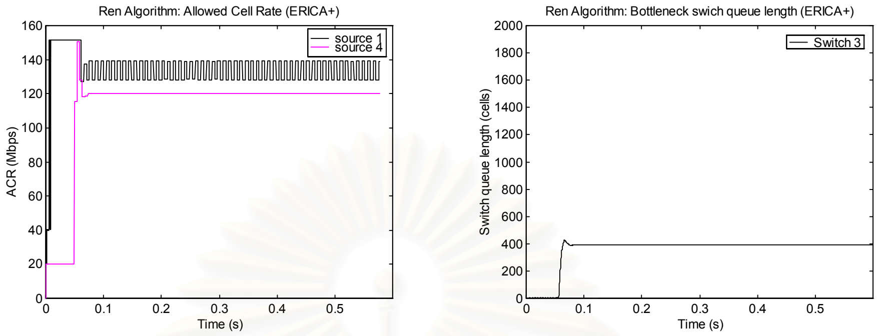
(a) Allowed cell rate (b) Switch queue length

Figure 5.9: Results for point-to-multipoint chain scenario

[Fahmy Algorithm – FRACA]

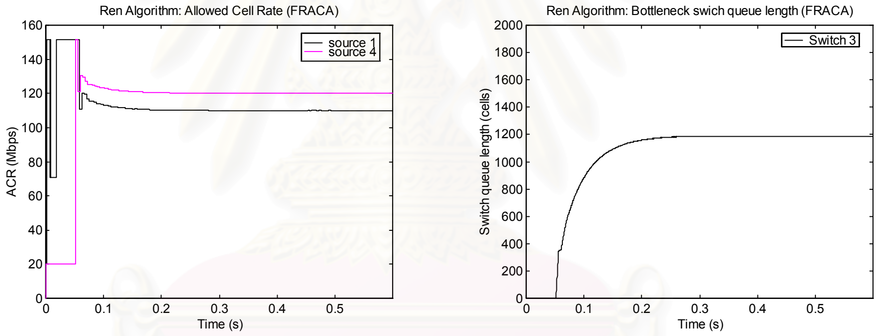
Figure 5.6 exhibits similar results to [20] that Ren algorithm over ERICA suffers from oscillation of allocated rates due to the consolidation noise. The unacceptable rate oscillation causes high buffer occupancy at the bottlenecked switch. In other words, the switch queue length could not be conducted to the desired level regards the queue control objective of the ERICA+ algorithm. However, the FRACA rate allocation algorithm significantly improves performance of Ren algorithm. Figure 5.7 illustrates the fast transient response while complete eliminating consolidation noise. In addition, at steady state the switch queue level is conducted to the desired level. For Fahmy algorithm over both FRACA and ERICA+ Figure 5.8 and 5.9, there is no significantly difference except that the response and convergence time of the FRACA algorithm is faster than ERICA+. The faster response can be obvious from the peak level of the switch queue and the time taken in order to conduct the queue to the designed level.

Figure 5.10 – 5.13 shows the results of point-to-multipoint chain scenario with MCR guarantee and bottlenecked source for Ren algorithm over GWF ERICA+, Ren algorithm over FRACA, Fahmy algorithm over GWF ERICA+ and Fahmy algorithm over FRACA, respectively.



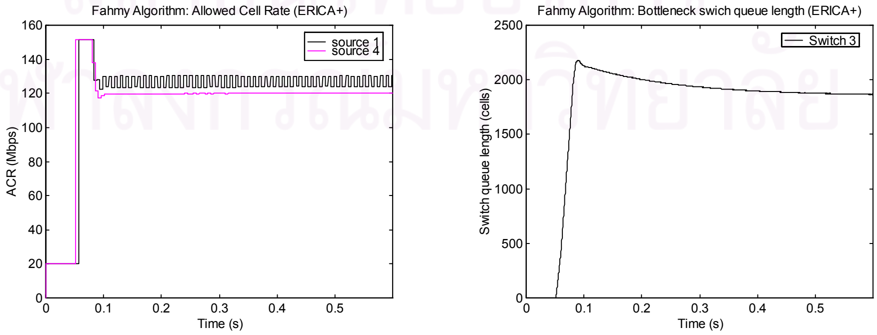
(a) Allowed cell rate (b) Switch queue length

Figure 5.10: Results for point-to-multipoint chain scenario with MCR guarantee and source bottleneck [Ren Algorithm – GWF ERICA+]



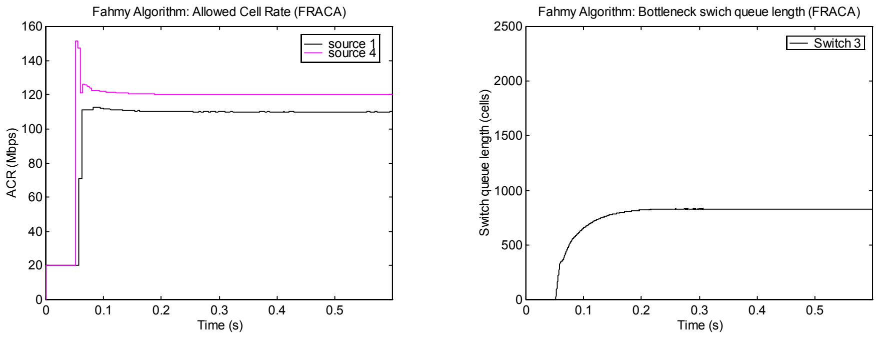
(a) Allowed cell rate (b) Switch queue length

Figure 5.11: Results for point-to-multipoint chain scenario with MCR guarantee and source bottleneck [Ren Algorithm – FRACA]



(a) Allowed cell rate (b) Switch queue length

Figure 5.12: Results for point-to-multipoint chain scenario with MCR guarantee and source bottleneck [Fahmy Algorithm – GWF ERICA+]



(a) Allowed cell rate (b) Switch queue length

Figure 5.13: Results for point-to-multipoint chain scenario with MCR guarantee and source bottleneck [Fahmy Algorithm – FRACA]

Figure 5.10 and 5.12 indicate that the allocated rates of source 1 by the GWF ERICA+ algorithm with Ren and Fahmy consolidation algorithms could not converge to the expected rates. The allocated rate oscillates around 130.0 Mbps. instead of convergence to 110.0 Mbps. Working along with Fahmy consolidation algorithm, there is no consolidation noise due to the consolidation methodology. The error is caused by the use of CCR field in the rate allocation process of the GWF ERICA+ algorithm, i.e. the CCR field of source 1 does not reflect the actual transmit rate. Hence, the rate allocation algorithm in spite of the consolidation algorithm mainly affects the performance of point-to-multipoint ABR connections. Figure 5.11 and 5.13 indicate that working along with the non-zero MCR guarantee FRACA algorithm, both Ren and Fahmy consolidation algorithm converge to the expected rates with controllable switch queue level.

5.4 Summary

This chapter has examined the effect of explicit rate allocation algorithms to the performance of point-to-multipoint ABR connections. The purposed FRACA is used as a sample switch algorithm in order to compare the results, in term of allocated rates and switch queue level, with the ERICA+ and GWF ERICA+. It is obvious from the results that working with the FRACA algorithm a simple consolidation algorithm, i.e. Ren algorithm, performs well compared to the ERICA+ and GWF ERICA+. In addition, although using a modern consolidation algorithm, i.e. Fahmy algorithm, over ERICA+ or GWF ERICA+ the consolidation noise is completely eliminated but the response and convergence time is longer than running over FRACA. Moreover, this dissertation pioneers that the source traffic characteristics also affect the performance of point-to-multipoint ABR connections that use the ERICA+ or GWF ERICA+ as a switch algorithm. The limitations are caused by the computation steps in the ERICA+ and GWF ERICA+ that use the CCR field in RM cell, which may not reflect the source actual transmit rate.

CHAPTER 6

CONCLUSTION AND FUTURE WORK

As a packet-switching technique, ATM has been proposed as the enabling technology for future high-speed networks. The desire to support a diverse range of traffic types has led to the development of several user traffic classes. Basically, data traffic is highly unpredictable and usually more delay tolerant than other traffic types. In light of these requirements, the standard bodies have defined the ABR service class to maximize network bandwidth utilization. The design of robust ABR schemes, capable of functioning in a wide range of network conditions, is therefore essential.

Although various ABR congestion control proposals have been published previously, many important issues still remain outstanding, especially some necessary provisions for functionality in dynamic environments are missing. These include features for handling bursty sources, scalability in complex network scenarios, supporting a large number of traversing ABR sessions and system propagation delay [37]. In addition, most of the existing schemes lack clear extensions to handle MCR guarantees among connections.

6.1 Conclusion

The Fast Rate Allocation Congestion Avoidance (FRACA) scheme, which is fully compatible with ATMF traffic management 4.0 [1], is introduced to solve deficiencies in the existing ABR flow control proposals. The main advantage of the FRACA scheme is that the performance is tolerant to network conditions such as number of traversing ABR sessions, network scenario, propagation delay and aggregate traffic characteristics. The allocated rates in all evaluated scenarios converge to the expected max-min rates. Unlike other existing algorithms, ERICA+ and E-FMMRA, which the algorithm parameters have to be re-turned when network conditions change. This may cause both algorithms function not efficiently in the real working environments. In addition, the MCR-plus-equal-share fairness criterion is implemented to properly handle bandwidth partitioning beyond the MCR guarantees. The performance of the MCR guarantee FRACA scheme also works better than GWF ERICA+ in terms of the convergence time and the tolerance to the change in network conditions. Finally, this dissertation pioneers the effects of the rate allocation algorithm on the performance of point-to-multipoint ABR service. Working along with the proposed FRACA scheme, a simple consolidation algorithm performs an acceptable behavior, i.e. the fast convergence time while the consolidation noise is eliminated. Consequently, the implementation complexity of the point-to-multipoint connections in an ATM switch is significantly reduced.

6.2 Future work

In this dissertation, a comprehensive strategy for a rate allocation in the ABR services in ATM networks is presented and evaluated by means of extensive simulation and some analysis. However, there are various important issues relating to the proposed algorithm, which deserve further investigation. Those are:

- Currently, the ABR service is being actively implemented and faces some interesting cost-performance tradeoff questions. To make the service more attractive in the cost-performance tradeoff is to demonstrate that a large class of application, i.e. variable quality voice and video, can be made to run over the service with the pre-promised QoS.
- Typically, the ABR provides control only up to the edge of the ATM networks. However, it is possible that the edge switch can use the ABR feedback information to pace the TCP control, i.e. TCP congestion control windows. This will carry the benefits of ABR to applications.
- For ABR service over Asymmetric Digital Subscriber Line (ADSL), the performance of the proposed FRACA algorithm still needs future investigation. The algorithm may enhance the utilization of the existing telephone line. Therefore, it will support the growth of Internet users in Thailand.

REFERENCES

[1] The ATM Forum, "Traffic Management Specification version 4.0," AF-TM-0056.000, April 1996.

[2] Arulambalam and N. Ansari, "An Intelligent Explicit Rate Control Algorithm for ABR Service in ATM networks," Proceeding of IEEE International Conference on Communications, 1997.

[3] Chiussi, Arulambalam, Ye Xia, and 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," IEEE/ACM Transaction on Networking, February 2000.

[5] Shiv Kalyanaraman, Bobby Vandalore, Raj Jain, Rohit Goyal and Sonia Fahmy, "Performance of TCP over ABR with Long-Range Dependent VBR Background Traffic over Terrestrial and Satellite ATM Networks," Proceeding of 23rd Annual Conference on Local Computer Networks, October 1998.

[6] Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, Shiv Kalyanaraman, S. Kota and P. Samudra, "Feedback consolidation algorithms for ABR point-to-multipoint connections," Proceeding of IEEE INFOCOM, volume 3, pp.1004-1013, March 1998.

[7] T. Jaruvitayakovit, N. Rangsinoppamas and P. Prapinmongkolkarn: "Improvement of ERICA+ Scheme using Adaptive Average Interval Algorithm," Proceeding of the Applied Telecommunications Symposium (ATS'99), San Diego, USA., April 1999.

[8] Robert J. Simcoe, "Test configurations for fairness and other tests," ATM Forum/94-0557, July 1994.

[9] Bobby Vandalore, Raj Jain, Rohit Goyal and Sonia Fahmy, “Design and Analysis of Queue Control Function for Switch Scheme,” ATM Forum/97-1087, December 1997.

[10] M. Laubach and J. Halpern, “Classical IP and ARP over ATM,” IETF RFC2225, April 1998.

[11] T. Jaruvitayakovit, N. Rangsinoppamas, B. Tansuthepverawongse and P. Prapinmongkolkarn, “A novel efficient and robust explicit rate for ABR service in ATM networks,” Proceeding of the Fifth IFIP Conference on Intelligence in Networks, Asian Institute of Technology, Thailand, November 1999.

[12] Arulambalam, X. Chen and N. Ansari, “Allocating Fair Rates for Available Bit Rate Service in ATM Networks,” IEEE communication magazine, pp.92-100, November 1996.

[13] T. Jaruvitayakovit, N. Rangsinoppamas and P. Prapinmongkolkarn, “Performance Re-evaluation of point to multipoint ABR Service in ATM Networks,” Proceeding of International Symposium on Intelligent Signal Processing and Communication Systems, Hawaii, USA., November 2000.

[14] Raj Jain, “Congestion control and traffic management in ATM networks: Recent advances and a survey,” Computer Networks and ISDN Systems, Vol. 28, No. 13, October 1996, pp.1723-1738

[15] Shiv Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, Fang Lu and Saragur Srinidhi, “Performance of TCP/IP over ABR,” Proceeding of Globecom, November 1996

[16] Yiwei Thomas Hou, Henry H.-Y. Tzeng and Shivendra S. Panwar, “A generalized Max-Min Rate Allocation Policy and Its Distributed Implementation Using the ABR Flow Control Mechanism,” Proceeding of INFOCOM, 1998

[17] Yiwei Thomas Hou, Henry H.-Y. Tzeng and Shivendra S. Panwar, “A Simple ABR Switch Algorithm for the Weighted Max-Min Fairness Policy,” Proceeding of IEEE ATM Workshop, 1997.

[18] Y.H. Long, T.K. Ho and A.B. Rad, “Explicit Rate allocation algorithm of generalized max-min fairness for ATM ABR services,” IEE Electronics Letters, April 1999.

[19] Bobby Vandalore, Sonia Fahmy, Raj Jain, Rohit Goyal and Mukul Goyal, “General Weighted Fairness and its Support in Explicit Rate Switch Algorithms,” Journal of Computer Communications, January 2000.

[20] Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, Shiv Kalyanaraman, S. Kota and P. Samudra, “Design and Evaluation of Feedback consolidation for ABR point-to-multipoint connections in ATM Networks,” Journal of Computer Communications, Vol. 22, Issue 12, pp. 1085-1103.

[21] Sonia Fahmy, Raj Jain, Shiv Kalyanaraman, Rohit Goyal and Bobby Vandalore, “On determining the fair bandwidth share for ABR connections in ATM networks.” Proceeding of IEEE International Conference on Communications, Atlanta, Vol. 3, pp. 1485-1491, June 1998.

[22] N. Ghani and J. W. Mark, “Enhanced Distributed Explicit Rate Allocation for ABR service in ATM Networks,” IEEE/ACM Transaction on Networking, February 2000.

[23] Shiv Kalyanaraman, “Traffic management for the available bit rate ABR service in Asynchronous Transfer Mode (ATM) networks,” Ph.D. dissertation, The Ohio State University, 1997.

[24] Rohit Goyal, “Traffic management for TCP/IP over Asynchronous Transfer Mode (ATM) networks,” Ph.D. dissertation, The Ohio State University, 1999.

[25] Sonia Fahmy, "Traffic Management for Point-to-Point and Multipoint Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks." Ph.D. dissertation, The Ohio State University, 1999.

[26] Bobby Vandalore, "Traffic Management to Enhance Quality of Service (QoS) of Multimedia over Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks," Ph.D. dissertation, The Ohio State University, 2000.

[27] H-Y Tzeng and K-Y Siu, "On max-min fair congestion control for multicast ABR service in ATM," IEEE Journal on Selected Areas in Communications, Vol. 15, no. 3, April 1997.

[28] You-Ze Cho, Sang-Min Lee and Myeong-Young Lee, "An efficient rate-based algorithm for point-to-multipoint ABR service," proceeding of IEEE GLOBECOM, pp. 790-795, November 1997.

[29] Hung-Shiun Alex Chen and Klara Nahrstedt, "Feedback consolidation and timeout algorithms for point-to-multipoint ABR service," Proceeding of IEEE International Conference on Communications, Vancouver, June 1999.

[30] Dong-Ho Kim and You-Ze Cho, "A scalable consolidation algorithm for point-to-multipoint ABR flow control in ATM networks," Proceeding of IEEE International Conference on Communications, Vancouver, June 1999.

[31] W. Ren, K-Y Siu and H. Suzuki, "On the performance of congestion control algorithms for multicast ABR service in ATM," Proceeding of IEEE ATM Workshop, San Francisco, August 1996.

[32] R. Onurval, "Asynchronous Transfer Mode Networks: Performance Issues," Artech House, 1994.

[33] D. Bersekas and R. Gallager, "Data Networks," Prentice Hall, 1992.

[34] V. Jacobson, "Congestion avoidance and control," Proceeding of SIGCOMM 1988.

[35] W. Stevens, “TCP/IP Illustrated Volume 1,” Addison Wesley, 1994.

[36] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, “An Architecture for Differentiated Services,” IETF RFC 2475, 1998.

[37] H. Ohsaki, M.Murata, H.Suzuki, C. Ikeda, H. Miyahara, “Rate-based congestion control for ATM networks,” Computer Communication Review, Vol. 25, No. 2, April 1995, pp. 60-72.

[38] H. Ohsaki, M.Murata, H.Suzuki, C. Ikeda, H. Miyahara, “Performance evaluation of rate-based congestion control algorithms in multimedia for ATM networks,” Proceeding of IEEE Globecom, Singapore, November 1995, Vol. 2, pp. 1243-1248

[39] H. Saito, “Performance issues in public ABR service,” IEEE Communication Magazine, Vol. 34, No. 11, November 1996, pp. 40-48

[40] L. Roberts, “Enhance PRCA (Proportional Rate Control Algorithm),” AF-TM 94-0916, September, 1994.

[41] P. Newman, “Traffic management for ATM Local Area Network,” IEEE Communications Magazine, August 1994, pp. 44-50.

[42] M. Baumann, “YATS – Yet Another Tiny Simulator user’s manual version 0.3,” http://www.ifn.et.tu-dresden.de/TK/yats/yats.html, Communication Laboratory, Dresden University of Technology, 1996.

[43] T. Jaruvitayakovit, N. Rangsinoppamas and P. Prapinmongkolkarn, ‘Analysis and design of a stable congestion avoidance algorithm for ABR service in ATM networks’. Accepted to be published in IEICE Transaction on Communications, April 2002.

[44] T. Jaruvitayakovit and P. Prapinmongkolkarn, 'Performance analysis of congestion avoidance algorithms with MCR guarantee for ABR service in ATM networks'. Submitted for publication, 2002.

APPENDICES

APPENDIX A
SOURCE, DESTINATION AND SWITCH RULES

This appendix provides the precise source and destination behavior verbatim from the ATM Forum's Traffic Management 4.0 specification [1]. All table, section, and other references in this appendix refer to those in the TM specification.

A.1 Source Behavior

The following items define the source behavior for CLP=0 and CLP=1 cell streams of a connection. By convention, the CLP=0 stream is referred to as in-rate, and the CLP=1 stream is referred to as out-of-rate. Data cells shall not be sent with CLP=1.

1. The value of ACR shall never exceed PCR, nor shall it ever be less than MCR. The source shall never send in-rate cells at a rate exceeding ACR. The source may always send in-rate cells at a rate less than or equal to ACR.
2. Before a source sends the first cell after connection setup, it shall set ACR to at most ICR. The first in-rate cell shall be a forward RM-cell.
3. After the first in-rate forward RM-cell, in-rate cells shall be sent in the following order:

   a) The next in-rate cell shall be a forward RM cell if and only if, since the last in-rate forward RM-cell was sent, either:

   i) at least Mrm in-rate cells have been sent and at least Trm time has elapsed, or

   ii) Nrm -1 in-rate cells have been sent.

   b) The next in-rate cell shall be a backward RM-cell if condition (a) above is not met, if a backward RM cell is waiting for transmission, and if either:

   i) no in-rate backward RM-cell has been sent since the last in-rate forward RM-cell, or

   ii) no data cell is waiting for transmission.

   c) The next in-rate cell sent shall be a data cell if neither condition (a) nor condition (b) is met, and if a data cell is waiting for transmission.

4. Cells sent in accordance with source behaviors #1,#2, and #3 shall have CLP=0.
5. Before sending a forward in-rate RM cell, if ACR > ICR and the time T that has elapsed since the last in-rate forward RM-cell was sent is greater than ADTF, then ACR shall be reduced to ICR.
6. Before sending an in-rate forward RM cell, and following behavior #5 above, if at least CRM in-rate forward RM-cells have been sent since the last backward RM-cell with BN=0 was received, then ACR shall be reduced by at least ACR×CDF, unless that reduction would result in a rate below MCR, in which case ACR shall be set to MCR.
7. After following behaviors #5 and #6 above, the ACR value shall be placed in the CCR field of the outgoing forward RM-cell, but only in-rate cells sent after the outgoing forward RM-cell need to follow the new rate.
8. When a backward RM-cell (in-rate or out-of-rate) is received with CI=1, then ACR shall be reduced by at least ACR×RDF, unless that reduction would result in a rate below MCR, in which case ACR shall be set to MCR. If the backward RM-cell has both CI=0 and NI=0, then the ACR may be increased by no more than RIF×PCR, to a rate not greater than PCR. If the backward RM-cell has NI=1, the ACR shall not be increased.
9. When a backward RM-cell (in-rate or out-of-rate) is received, and after ACR is adjusted according to source behavior #8, ACR is set to at most the minimum of ACR as computed in source behavior #8, and the ER _eld, but no lower than MCR.
10. When generating a forward RM-cell, the source shall assign values to the various RM-cell fields as specified for source-generated cells in Table 5-4.
11. Forward RM-cells may be sent out-of-rate (i.e., not conforming to the current ACR). Out-of-rate forward RM-cells shall not be sent at a rate greater than TCR.
12. A source shall reset EFCI on every data cell it sends.
13. The source may implement a use-it-or-lose-it policy to reduce its ACR to a value which approximated the actual cell transmission rate. Use-it-or-lose-it policies are discussed in Appendix I.8.

Notes:

1. In-rate forward and backward RM-cells are included in the source rate allocated to a connection.
2. The source is responsible for handling congestion within its scheduler in a fair manner. This congestion occurs when the sum of the rates to be scheduled exceeds the output rate of the scheduler. The method for handling local congestion is implementation specific.

A.2 Destination Behavior

The following items define the destination behavior for CLP=0 and CLP=1 cell streams of a connection. By convention, the CLP=0 stream is referred to as in-rate, and the CLP=1 stream is referred to as out-of-rate.

1. When a data cell is received, its EFCI indicator is saved as the EFCI state of the connection.
2. On receiving a forward RM-cell, the destination shall turn around the cell to return to the source. The DIR bit in the RM-cell shall be changed from "forward" to "backward," BN shall be set to zero, and CCR, MCR, ER, CI, and NI fields in the RM-cell shall be unchanged except:

   a) If the saved EFCI state is set, then the destination shall set CI=1 in the RM cell, and the saved EFCI state shall be reset. It is preferred that this step is performed as close to the transmission time as possible;

   b) The destination (having internal congestion) may reduce ER to whatever rate it can support and/or set CI=1 or NI=1. A destination shall either set the QL and SN fields to zero, preserve these fields, or set them in accordance with ITU-T Recommendation I.371-draft. The octets defined in Table 5-4 as reserved may be set to 6A (hexadecimal) or left unchanged. The bits defined as reserved in Table 5-4 for octet 7 may be set to zero or left unchanged. The remaining fields shall be set in accordance with Section 5.10.3.1 (Note that this does not preclude looping fields back from the received RM cell).
3. If a forward RM-cell is received by the destination while another turned-around RM-cell (on the same connection) is scheduled for in-rate transmission:

a) It is recommended that the contents of the old cell are overwritten by the contents of the new cell;

b) It is recommended that the old cell (after possibly having been overwritten) shall be sent out-of-rate; alternatively the old cell may be discarded or remain scheduled for in-rate transmission;

c) It is required that the new cell be scheduled for in-rate transmission.

4. Regardless of the alternatives chosen in destination behavior #3, the contents of the older cell shall not be transmitted after the contents of a newer cell have been transmitted.
5. A destination may generate a backward RM-cell without having received a forward RM-cell. The rate of the backward RM-cells (including both in-rate and out-of-rate) shall be limited to 10 cells/second, per connection. When a destination generated an RM-cell, it shall set either CI=1 or NI=1, shall set BN=1, and shall set the direction to backward. The destination shall assign values to the various RM-cell fields as specified for destination generated cells in Table 5-4.
6. When a forward RM-cell with CLP=1 is turned around it may be sent in-rate (with CLP=0) or out-of-rate (with CLP=1)

Notes

1. "Turn around" designates a destination process of transmitting a backward RM-cell in response to having received a forward RM-cell.

2. It is recommended to turn around as many RM-cells as possible to minimize turn-around delay, first by using in-rate opportunities and then by using out-of-rate opportunities as available. Issues regarding turning RM-cells around are discussed in Appendix I.7.

A.3 Switch Behavior

The following items define the switch behavior for CLP=0 and CLP=1 cell streams of a connection. By convention, the CLP=0 stream is referred to as in-rate, and the CLP=1 stream is referred to as out-of-rate. Data cells shall not be sent with CLP=1.

1. A switch shall implement at least one of the following methods to control congestion at queuing points:

a) EFCI marking: The switch may set the EFCI state in the data cell headers;

b) Relative Rate Marking: The switch may set CI=1 or NI=1 in forward and/or backward RM-cells;

c) Explicit Rate Marking: The switch may reduce the ER field of forward and/or backward RM-cells (Explicit Rate Marking);

d) VS/VD Control: The switch may segment the ABR control loop using a virtual source and destination.

2. A switch may generate a backward RM-cell. The rate of these backward RM cells (including both in-rate and out-of-rate) shall be limited to 10 cells/second, per connections. When a switch generates an RM-cell it shall set either CI=1 or NI=1, shall set BN=1, and shall set the direction to backward. The switch shall assign values to the various RM-cell fields as specified for switch-generated cells in Table 5-4.
3. RM-cells may be transmitted out of sequence with respect to data cells. Sequence integrity within the RM-cell stream must be maintained.
4. For RM-cells that transit a switch (i.e., are received and then forwarded), the values of the various fields before the CRC-10 shall be unchanged except:

   a) CI,NI and ER may be modified as noted in #1 above

   b) RA, QL and SN shall be set in accordance with ITU-T Recommendation I.371-draft.
5. The switch may implement a use-it-or-lose it policy to reduce an ACR to a value which approximates the actual cell transmission rate from the source. Use-it-or-lose-it policies are discussed in Appendix I.8.

Notes

1. A switch queuing point is a point of resource contention where cells may be potentially delayed or lost. A switch may contain multiple queuing points.
2. Some example switch mechanisms are presented in Appendix I.5.
3. The implications of combinations of the above methods is beyond the scope of this specification.

APPENDIX B

TCP/IP OVER ABR SERVICE IN ATM NETWORKS

Most of the traffic on the Internet today is data traffic in the sense that they are bursty and relatively delay insensitive. ABR service class has been developed specifically to support data applications, it is important to investigate the performance of dominant internet applications like file transfer and world wide web (which use TCP/IP) running over ABR. In this section, we provide a detailed study of the dynamics and performance of TCP/IP over an explicit rate congestion avoidance scheme in ABR networks.

Transmission Control Protocol (TCP) is an end-to-end congestion control mechanism that is used to guarantee reliability for data transmission. Whereas Internet Protocol (IP) is widely used for addressing in existing Internet. Mostly existing data transfer applications employ TCP/IP as the set of protocols for transmission control. Hence, the performance of a stable congestion avoidance algorithm should not be affected by TCP/IP applications. This section gives an overview of TCP protocol and discusses the behavior of running TCP over rate allocation algorithm in ABR service. The protocol layers for transporting TCP/IP applications over ATM networks is shown in Figure B.1. User data is divided into TCP segments that have length as the negotiated Maximum Segment Size (MSS). The TCP layer attaches a 20 byte header to the segment and passes it to the IP layer which also attaches 20 byte IP header. The IP packet is encapsulated over ATM networks according to the specification defined in Request For Comment 2225 (RFC2225) [10]. The layer attaches 8 bytes of Logical Link Control (LLC) header to the IP packet before sending the packet to ATM Adaptation Layer 5 (AAL5) layer. The AAL5 layer includes another 8 byte trailer to form a AAL5 frame. The AAL5 frame is encapsulated to be ATM cell payloads of size 48 bytes each and attached 5 bytes ATM cell header to each payload by ATM layer. Figure B.2 shows the segmentation procedures. At the destination end system, each layer strips off the appropriate header in the reverse procedures as the source end system before reassembly to construct the original user data frame.

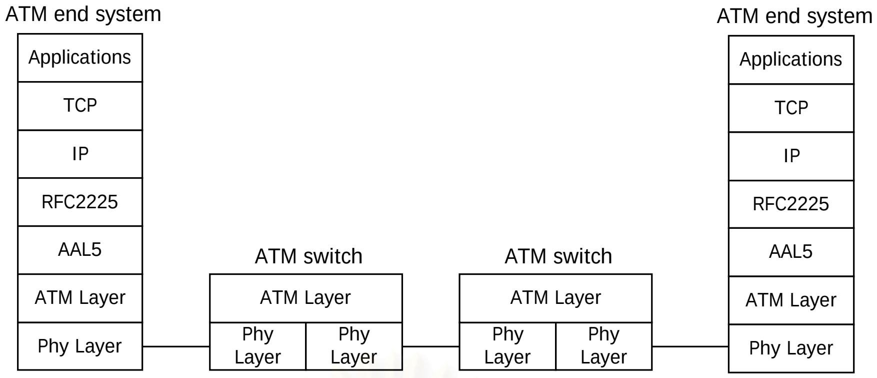


Figure B.1: TCP/IP over ATM protocol layers

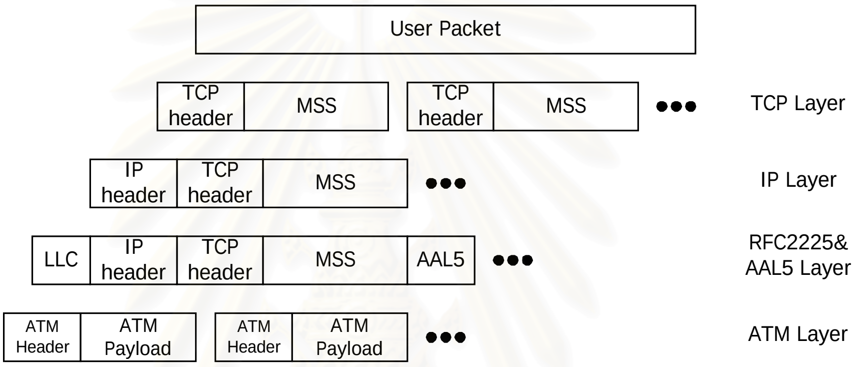
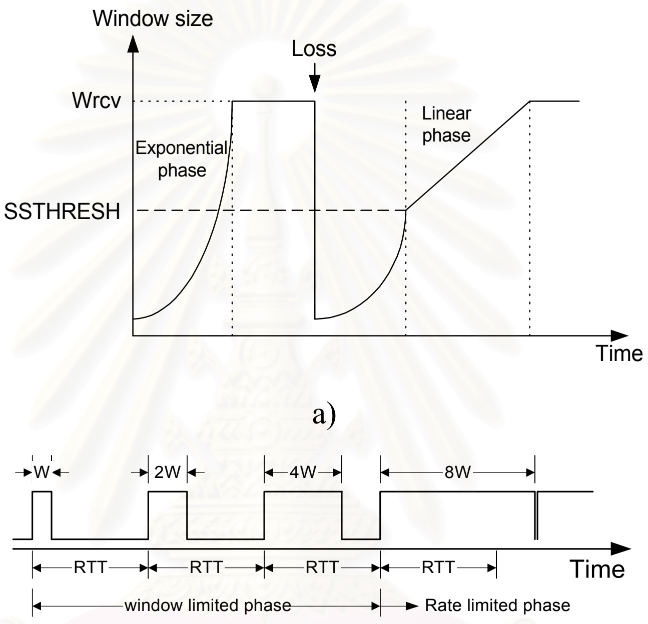


Figure B.2: Segmentation of TCP packet to AAL5 cells

TCP sessions use flow control window to limit the number of packets that the sources send into the network [35]. The sender window size is the minimum of the receiver window (Wrcv) and the congestion window (CWND). The key TCP congestion mechanism is call “Slow Start”. When a new session is established, the CWND is initialized to one segment (packet size). Each time an Acknowledge (ACK) is received, the CWND is increased by one segment. The sender can transmit up to the minimum of the CWND and Wrcv. This phase is also called as “exponential increase phase” since the window when plotted as a function of time increases exponentially.

When the network congests, packets are randomly discarded. There are two indications of packet loss: a timeout occurring and the receipt of duplicate ACK (if the fast retransmission option is implemented). After packet loss detection, the source set the Slow Start Threshold (SSTHRESH) at half the current window size (but at least two segments). In addition, if the congestion is indicated by a timeout, CWND is set to one segment. The source retransmits the lost packet and increases the CWND by one every time a packet is acknowledged. This continues until the CWND is equal to SSTHRESH. After that CWND is increased by 1/CWND for every acknowledged packet. This phase is called “Congestion Avoidance” due to the behavior. This phase is also called “linear increase phase” since the window graph as a function of time is a straight line. Figure B.3 illustrates behavior of exponential-linear phase and congestion windows size of TCP.



a)

b)

Figure B.3:

a) TCP congestion window size vs time showing Exponential and Linear phase.

b) Time scale of TCP congestion window size vs Round Trip Time.

When a new TCP session starts sending data, the file transfer data is bottlenecked by the TCP congestion window size (not by ABR control loop). In this state, a TCP session is said to be “window-limited” [15, 23]. On the other hand, the exponential growth of CWND continues until window size is greater than the source’s sending rate, then it will be limited by the sending rate. The transferred data is bottlenecked by the ABR source rate (not by TCP congestion window size). In this case, a TCP session is said to be “rate-limited”.

Most of the previous researches assume that the sources are persistent, i.e. send data all the time, which only matches with rate-limited condition in running TCP over ABR service. Unlike the previous work, we also take into account the effects of window-limited (slow start phase) in TCP for designing the congestion avoidance algorithm. From Figure B.3 b) during window-limited phase, the source will become idle after sending a number of data cells according to congestion window size. Then the CCR field of RM cell (in ABR control loop) does not reflect the actual transmit rate. In this scenario, the congestion avoidance algorithm that is designed for persistent sources or uses CCR field in RM cell for rate allocation will perform worse than the algorithm that employs averaging of the aggregate traffic and sets average interval long enough to accurately average the aggregate traffic.

APPENDIX C

THE FRACA FLOWCHART

This appendix presents flowcharts to describe the proposed FRACA algorithm. The following names are used to identify the flowcharts:

Flowchart 1: Flowchart of the FRACA algorithm named as Figure C.1.

Flowchart 2: Flowchart for computing the load factor at the end of averaging interval named as Figure C.2.

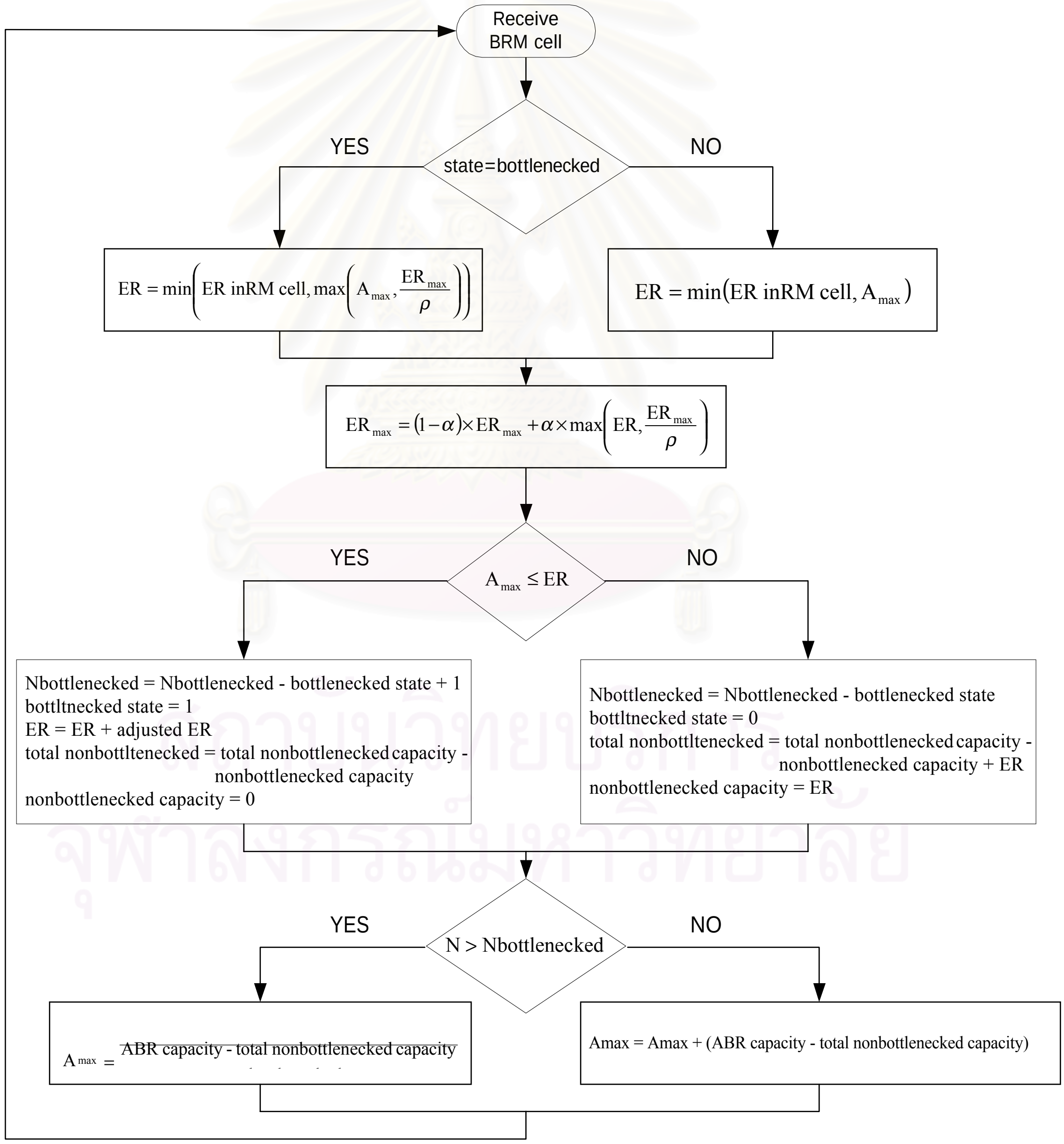

Receive BRM cell
state=bottlenecked
YES
NO
$ER = \min\left(ER \text{ inRM cell}, \max\left(A_{max}, \frac{ER_{max}}{\rho}\right)\right)$
$ER = \min(ER \text{ inRM cell}, A_{max})$
$ER_{max} = (1-\alpha)\times ER_{max} + \alpha\times\max\left(ER, \frac{ER_{max}}{\rho}\right)$
$A_{max} \le ER$
YES
NO
Nbottlenecked = Nbottlenecked - bottlenecked state + 1
bottltnecked state = 1
ER = ER + adjusted ER
total nonbottltenecked = total nonbottlenecked capacity - nonbottlenecked capacity
nonbottlenecked capacity = 0
Nbottlenecked = Nbottlenecked - bottlenecked state
bottltnecked state = 0
total nonbottltenecked = total nonbottlenecked capacity - nonbottlenecked capacity + ER
nonbottlenecked capacity = ER
N > Nbottlenecked
YES
NO
$A_{max} =$ ABR capacity - total nonbottlenecked capacity
Amax = Amax + (ABR capacity - total nonbottlenecked capacity)

Figure C.1: Flowchart of the FRACA algorithm

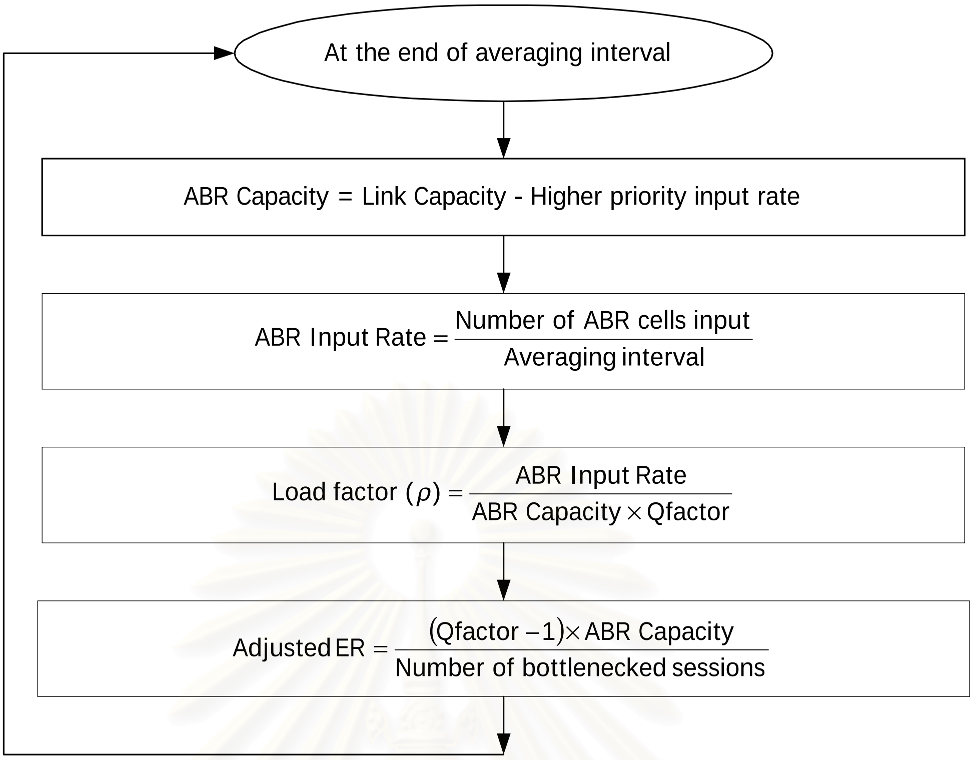

At the end of averaging interval
ABR Capacity = Link Capacity - Higher priority input rate
$\text{ABR Input Rate} = \frac{\text{Number of ABR cells input}}{\text{Averaging interval}}$
$\text{Load factor } (\rho) = \frac{\text{ABR Input Rate}}{\text{ABR Capacity} \times \text{Qfactor}}$
$\text{Adjusted ER} = \frac{(\text{Qfactor} - 1) \times \text{ABR Capacity}}{\text{Number of bottlenecked sessions}}$

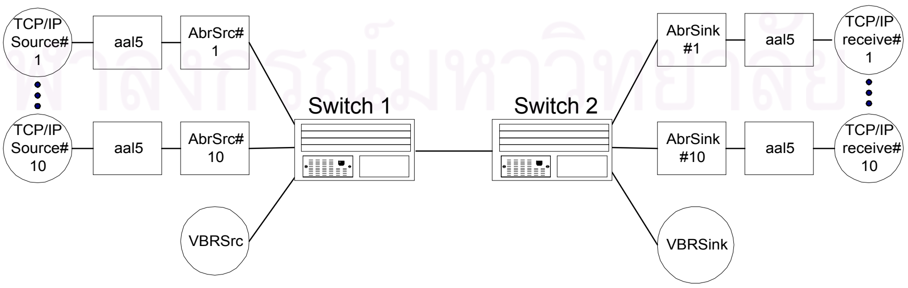
Figure C.2: Flowchart for computing the load factor at the end of averaging interval

APPENDIX D

THE AVERAGING FACTOR

In this appendix, we study the effect of the averaging factor (α) that used in the rate allocation process, i.e. equation (3-3). Normally, the averaging factor is used for weighting the ER_{max} in the last averaging interval and the computed ER_{max} in this averaging interval (ER_{max}/ρ). The lower value of the averaging factor causes the less effect of the computed ER_{max} in this averaging interval, compared with the ER_{max} in the last averaging interval. Remember that the ER_{max} is updated once for every connection in an averaging interval. Thus, the ER_{max} is updated *n*-time for an averaging interval where *n* is the number of the traversing connections that a switch receives their BRM cells during one averaging interval. Consequently, a too high value of the averaging factor causes the algorithm too sensitive to the aggregate traffic. On the other hand, a too low value of the averaging factor causes the slow response of the algorithm.

Let us use the simulation result to illustrate the effect of the averaging factor to the algorithm performance. The available bandwidth of the selected configuration should be varied over the time in order to evaluate the adaptation of the algorithm, as the effects of the averaging interval. The evaluated scenario is the 10 TCP sources plus VBR configuration (the same configuration as Figure 3.19). The mathematical mean and the standard deviation of the Allowed Cell Rate and the switch queue are the metrics used to evaluate the effects of the averaging factor to the algorithm performance. The configuration and parameter setting are shown in Figure D.1.



Parameter setting:

- TCP/IP : TCP sources MSS = 8 kbytes, TCP receiver window size = 64 kbytes.

- All ABR sources PCR = Link capacity = 149.76 Mbps., MCR = 0.0 Mbps., ICR = 5.0 Mbps.
- VBR sources are on-off sources, i.e. on-off time is 5 msec., aggregate VBR traffic is 120 Mbps during on period.
- All link propagation delay = 5 msec.

Figure D.1: 10 TCP sources plus VBR configuration

The results, mean and standard deviation of the Allowed Cell Rate and switch queue, of the 10 TCP sources plus VBR configuration are shown in Figure D.2 and D.3, respectively.

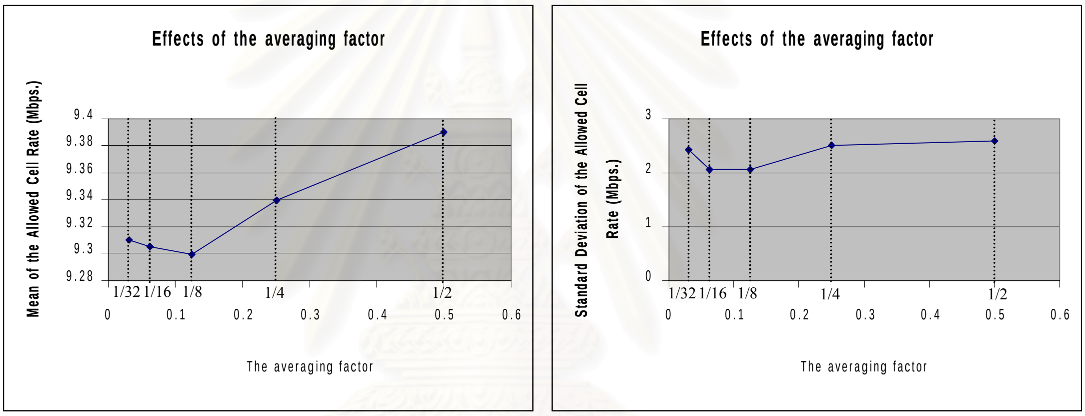


Figure D.2: Mean and standard deviation of the Allowed Cell Rate

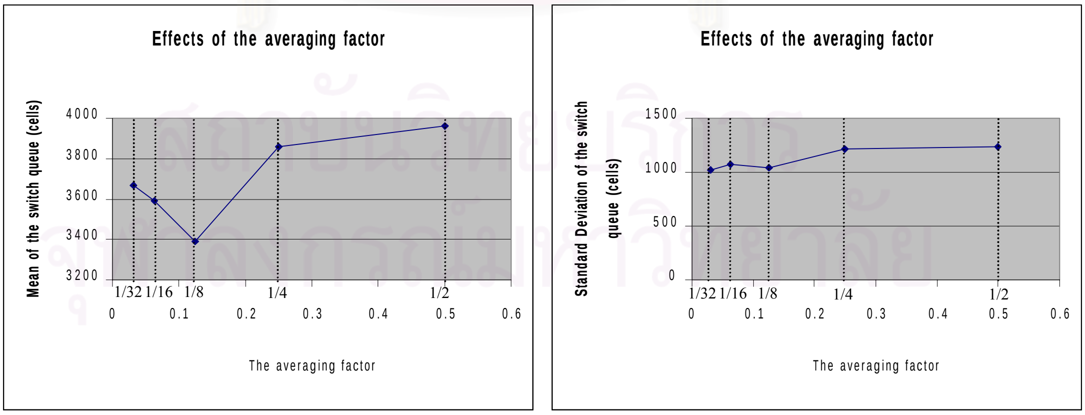


Figure D.3: Mean and standard deviation of the switch queue

In the configuration, the max-min fair rate for all sources should be (150 – 120/2) / 10 = 9 Mbps. From the results in Figure D.2, the mean of the Allowed Cell

Rate of a source is a little bit greater than the max-min fair rate for all values of the averaging factor. This is because there is some ABR cells buffered in the switch queue due to the queue control option. From the results, the higher value of the averaging factor, i.e. 1/4 and 1/2, causes the algorithm too sensitive to the aggregate traffics. Consequently, the mean and standard deviation of the Allowed Cell Rate and switch queue are higher than those of the lower value of the averaging factor. There is no significant difference between the results of the averaging factor of 1/8, 1/16 and 1/32. However, the mean of the switch queue of the averaging factor 1/8 is a little bit lower than those of the averaging factor 1/16 and 1/32, respectively. This is because the FRACA algorithm with the averaging factor of 1/8 can able to track down the aggregate traffics faster.

Therefore, this dissertation recommends the averaging factor 1/8 to be used with the proposed FRACA algorithm.

Biography

Mr. Tanun Jaruvitayakovit was born May 29, 1973 in Bangkok. He was graduated both Bachelor (Hons) and Master degree at Electrical Engineering Department, Chulalongkorn university in 1994 and 1996, respectively. From 1996 to 1997, he worked as a network engineer at Telesat Corporation, Co. Ltd. He has been working on his Doctor of Engineering in Chulalongkorn university since 1997. During the study, he received a scholarship from the National Science and Technology Development Agency (NSTDA) and the Thailand Research Fund (TRF).