Université d'Ottawa · University of Ottawa

Real-time Video Transmission over ABR Service using Rate Control Algorithm

By

Jianyuan Fan

A Thesis submitted to
The Faculty of Graduate Studies
University of Ottawa
In partial fulfillment of the requirements
for the degree of

Master of Applied Science
In Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa
Ottawa, Ontario, Canada

April 2002

0-612-76578-4

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Luis Orozco-Barbosa, for his guidance and advice throughout my graduate program.

I would also like to thank my wife Lingyun Wang for her support and understanding during my thesis work.

ABSTRACT

The ATM Available Bit Rate (ABR) service category was initially designed to support non-real-time applications. However it is possible for ABR to support real-time applications, such as multimedia applications, by taking advantage of the inherent traffic control capability of ABR and the adaptation ability of real-time applications. Using a proper switch traffic control algorithm, ABR queues in the switches can be kept within acceptable bounds and therefore a loose end-to-end delay bound can be guaranteed. FASTRAC is a simple, scalable, provably stable, explicit rate computation scheme for flow control in computer networks. The main objective of the present thesis is to study how to transmit real-time video over ABR service using FASTRAC rate control algorithm. Using OPNET simulation tool, we study the effect of Resource Management (RM) cell interval, control delay and stability factor. With the simulation experiments, we study the performance of the rate control algorithm when it supports real-time video applications using fixed weight allocation scheme. To overcome the problems with the fixed weight allocation scheme, we introduce a dynamic weight allocation scheme, allowing us to improve the rate allocation among all ABR connections. Our simulation results also show that the spare link capacity can be further utilized by transmitting non-real-time applications without affecting the QoS of the real-time video applications.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF ACRONYMS

ABR	Available Bit Rate
ACR	Allowed Cell Rate
ATM	Asynchronous Transfer Mode
BECN	Backward Explicit Congestion Notification
BRM	Backward Resource Management (cell)
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CI	Congestion Indication (bit)
CLR	Cell Loss Ratio
CTD	Cell Transfer Delay
DCT	Discrete Cosine Transformation
EFCI	Explicit Forward Congestion Indication
EPICA	Enhanced Proportional Rate Control Algorithm
ER	Explicit Rate
ERICA	Explicit Rate Indication for Congestion Avoidance
FASTRAC	FAST RAte Computation
FEC	Forward Error Correction
FRM	Forward Resource Management (cell)
GFR	Guaranteed Frame Rate

GOP	Group Of Picture
HDTV	High Definition TeleVision
ICR	Initial Cell Rate
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MPEG	Moving Picture Experts Group
NI	No Increase (bit)
OPNET	OPtimum NETwork performance
QoS	Quality of Service
PCR	Peak Cell Rate
RIF	Rate Increase Factor
RM	Resource Management (cell)
RTT	Round Trip Time
SCR	Sustainable Cell Rate
TCR	Tagged Cell Rate
UBR	Unspecified Bit Rate
VBR	Variable Bit Rate
VC	Virtual Circuit
VoD	Video on Demand

CHAPTER 1 INTRODUCTION

1.1 Transporting rt-Video over ABR Service

In ATM networks, real-time multimedia applications are currently supported by the ATM CBR and rt-VBR service categories. The Constant Bit Rate (CBR) service category provides a constant amount of bandwidth, specified by the Peak Cell Rate (PCR), to be continuously available to the application. The network guarantees that all cells generated by the source that conform to the agreed PCR will be delivered with the agreed QoS parameters, namely minimal Cell Loss Ratio (CLR), fixed bounded Cell Transfer Delay (maxCTD) and peak-to-peak Cell Delay Variation (peak-to-peak CDV). This means that once the CBR connection is established, the QoS provided by the network to the application (e.g. real-time video) can be guaranteed. However this mode of operation will result in the connection refusal if the network does not have enough bandwidth resource to meet the source requirement specified by traffic parameters and QoS parameters [1]. In other words, it sacrifices multiplexing gain defeating one of the dominant advantages of ATM. Furthermore, when transporting compressed real-time video, since a compressed video flow is bursty, CBR requires the use of a local buffer for rate smoothing. Buffer overflow or underflow is prevented by continuously adjusting the quantifying parameters and which result in a variable video quality [2].

The rt-VBR service is characterized by Peak Cell Rate (PCR) and its associated Cell Delay Variation Tolerance (CDVT) as well as a set of traffic parameters (Sustainable Cell Rate and Maximum Burst Size) that control the bursty nature of rt-VBR (e.g. real-time video). It delivers cells of conforming connections with fixed bounds of maximum Cell Transfer Delay (maxCTD) and peak-to-peak Cell Delay Variation (CDV). Cells that are delayed beyond the value specified by maxCTD are assumed to be of significantly reduced value to the application. VBR requires accurate prediction of the traffic during contract admission in order to provide a statistical multiplexing gain and support the required QoS. However, traffic description can be difficult because of the unpredictable nature of real-time video. It is then obvious that VBR service for real-time video is often not feasible without proper online traffic measurement and QoS renegotiation mechanism [3].

Although CBR and VBR can guarantee a given QoS service for real-time applications, it is difficult, sometimes infeasible, to establish connections for real-time applications due to insufficient network resources and the problems of CBR and VBR services mentioned above. In this sense, the ABR service offers a number of benefits despite its perceived complexity. First, ABR connections share available network capacity: they have access to the instantaneous capacity unused by higher priority (CBR/VBR) connections. This unused capacity would otherwise be wasted. Thus the ABR service can increase the network utilization without affecting the quality of other services. Second, the share of available capacity used by a single ABR connection is dynamic and varies between an agreed Minimum Cell rate (MCR) and Peak Cell Rate (PCR). Third, the network provides feedback to ABR sources so that ABR flows are limited to the available capacity. Finally, for ABR sources that adapt their transmission rate to the provided feedback from network, a low Cell Loss Ratio (CLR) is guaranteed.

Recently there has been a great interest on the support of real-time video using the Available Bit Rate (ABR) service class. Although ABR has been originally designed by the ATM Forum to support best-effort services and it does not have any delay guarantees, it is possible to support real-time applications with ABR service if properly dealing with the following three issues:

- Efficient and fair feedback control algorithm;
- End-system adaptation to the feedback information from the network;
- Minimum Cell Rate (MCR) guarantee for ABR users.

Using proper switch traffic control algorithm, ABR queues in the switches can be kept very small and therefore a loose end-to-end delay bound can be guaranteed. In general, any switch algorithm with fast transient response and queue control can loosely guarantee low delay through the switch [4]. The real-time compressed video applications are capable of dynamically adapting to available bandwidth of the network by adjusting the compression parameters and other adaptation techniques. While the use of the Minimum Cell Rate (MCR) of ABR guarantees a minimum acceptable quality of the transported real-time video and makes the queuing delay bounded. This feature also distinguishes real-time ABR from completely rate-adaptive video transporting such as those used in Internet video applications [5] [6] [7].

With various applications being transported over heterogeneous network environments, both the networks and the applications should adapt to each other. The network should try to adapt to allocate its resources to meet the QoS requirements of the applications while the application should try to adapt to the changing condition of the network. In the ABR service, with the explicit rate feedback mechanisms discussed above, it is possible for both the network and the application to adapt to each other dynamically. The network tries to dynamically allocate bandwidth according to the source requirement specified in the request Explicit Rate (ER) field of RM-cells, while the source application will try to adapt to the network changing condition described by the feedback explicit rate in the RM-cell's ER field.

1.2 Overview of Previous Work

Most of the research work in this area has focused on ABR flow control algorithm and source adaptation technique. First, how the switch properly provides feedback information about the network condition and how the end-system reacts efficiently to the feedback information are two key issues for ABR to support real-time applications such as Video-on-Demand (VoD). In order to support real time applications, the end-system and network should try to adapt to each other. Second, the ATM Forum Traffic Management Specification (Version 4.1) only specifies the criteria of the switch behavior. It gives considerable freedom to switch vendors in designing and developing rate allocation algorithms. This leaves enough room for open research on the switch flow control scheme aiming to meet the characteristics of scalability, fairness, stability, simplicity to implement and more. Third, although the ATM Forum has specified the source end-system behaviors, the specification only specifies Allowed Cell Rate (ACR) at the network level. There is no specification or even suggestion on how real-time application should adapt itself (e.g. QoS and stream rate) at higher layers to meet the network condition level in terms of Allowed Cell Rate (ACR).

1.2.1 Real-time Application Adaptation

While most research on the multimedia application adaptation methods is centered around scalable video and smoothing techniques, adaptation of multimedia applications can be done at several layers of the network protocol stack [8]. At the physical layer, for example, adaptive power control techniques can be used to mitigate variations in a wireless environment. At the data link layer, adaptive error correction techniques (e.g. adaptive Forward Error Correction (FEC)) can be used to protect the variations of channel transmission quality. At the network layer, dynamic re-routing and fair share re-allocating mechanisms can be used to avoid congestion. At the transport layer, dynamic re-negotiation of connection parameters or different protocols can be used for adaptation.

At the application layer, the application can adapt to the changes of the network conditions by using several techniques including hierarchical encoding, dynamic compression, bandwidth smoothing, adaptive synchronization and interpolation techniques. While there are various QoS adaptation architectures and techniques, following discussion will focus on application level adaptation techniques that can change bandwidth requirements of multimedia application in order to adapt to the available network bandwidth resources.

[9] presented a very good review and classification of the current adaptation techniques based on the level of the system architecture at which adaptation occurs (system, platform, application or user). It argues that while current systems tend to support adaptation at a subset of these levels, future systems will be required to adapt at all of these levels. Operating systems should also be designed to dynamically manage system resources, such as CPU, memory, I/O resources and network, so that user prospective QoS can be protected from the variation of the network conditions. It is beyond the scope of this thesis to discuss these issues in detail. The thesis will focus on how ABR switch and sources behave when transporting real-time video traffic.

1.2.2 ABR Flow Control Mechanism

An ABR switch provides feedback information to the end-system by setting the EFCI and CI/NI bit or by changing the ER field of RM-cell. However, there are no standards describing the evaluation procedure of the feedback information as a function of the switch load. In the case of an Explicit Rate marking method, there has not yet been specified the way to compute the ER field fairly according to the traffic load of the switch and the bandwidth requirement of the source.

A number of switch control mechanisms have been proposed and studied in the literature. Some of them are listed in the ATM TM4.1 specification as typical ABR control mechanisms, such as the EFCI binary feedback scheme, the Enhanced Proportional Rate Control Algorithm (EPRCA), the Explicit Rate Indication for Congestion Avoidance (ERICA) [10] [11] [12]. There are also a few studies specified on the problem of weighted fairness computation schemes in the literature [13] [14].

There have been a limited number of studies that addressed the problem of transporting real-time video with feedback control over ABR service category. [15] discusses transporting hierarchically compressed scalable video on demand (only assumed two levels of quality of video) with the Binary Rate marking ABR algorithm. [16] shows how ABR explicit rate feedback can be used to transport compressed real-time video. The video teleconference sources adapt to the required rate by modifying the quantization level used during compression. In [17], an algorithm for transporting smoothed compressed video over explicit rate networks is proposed. In this study, a small number of frames are stored at the source buffer and are used for smoothing the demand from the source to the network, thus helping achieve good multiplexing gain. It tries to maintain the quality of the video as long as possible before performing adaptive video encoding. A few studies also focused on multimedia application over multipoint ABR connections and ABR performance analysis and verification by simulation [2] [3] [4] [18]. Other related studies show the way to help making the real-time video application more adaptive and error-resilient by introducing mechanisms at different layers [19] [20] [21] [8] [22].

However, many of the proposed algorithms are complex, not scalable to high speeds and to the number of connections, and their stability is difficult to prove. It's also difficult to establish the effectiveness of most algorithms when the control loop round trip time is large. It is believed that most of the proposed algorithms will be difficult to use in public networks due to the longer round-trip times and greater number of multiplexed connections. Most of the proposed algorithms have not been entirely satisfactory for implementation and network deployment. The main drawback of most algorithms proposed in the literature is that they are based on heuristics and depend on a number of parameters that have to be set right and also on a number of measurements that are costly and difficult to perform. Further, these proposals do not provide any formal design methodology to ensure the stability of the control loop.

FASTRAC, a simple, scalable, provably stable, explicit rate computation scheme for flow control in computer networks is proposed in [23]. The proposed algorithm takes into account the control loop delay and is scalable to high-speeds and a large number of connections. The Rate control algorithm has the following important attributes: (1) the algorithm uses two primary measurements, the aggregate background (CBR/VBR) traffic rate and the aggregate ABR traffic rate. The algorithm does not require per-connection rate measurements or variables. (2) The explicit rate, i.e., the fair rate allocation, is computed every t_m seconds (typically in the range of 0.5 ~ 1 ms for an OC3 link). This greatly reduces the processing overhead of the switch and the computation can be done in software, while most of the proposed algorithm will do ABR computations in hardware. (3) The proposed algorithm achieves max-min fairness and supports MCR guarantees (non-zero MCR) in a nature way without any additional computation or information about bottleneck rates.

Using the discrete-time analysis, FASTRAC rate control algorithm has been proved to be stable and fair to all participating sources and configurable with respect to responsiveness. The simulation results also demonstrated the stability to scale, with speed, distance, number of sources while remaining robust, efficient, and fair under stressing and dynamic traffic conditions, and multiple propagation delay loops.

1.3 Thesis Objectives and Organization

FASTRAC, the rate control algorithm under consideration, has been proven to stable and scalable in previous study. The main objective of the thesis is to examine the behavior and performance of the algorithm when supporting greedy traffic and rt-video traffic.

The thesis aims to address the following three issues:

(1) What is the effect of the various factors and parameters, such as RM cell interval, control delay, stability factor and bottleneck position, on the behavior and performance of a system using the FASTRAC rate control;

(2) How is the performance of the system when ABR sources support greedy non-real-time traffic, real-time video traffic or both;

(3) How to improve the overall QoS of the ABR connections when supporting real-time video traffic;

(4) Based on the simulation and analysis, to provide guidelines to setup real-time video applications over ABR services using FASTRAC rate control algorithm.

The main contributions of the thesis include the follows. The experiment simulation in the previous study only considered supporting greedy traffic over ABR service using FASTRAC rate control algorithm. In the thesis we extend the study by evaluating the performance when it supports real-time video traffic. To solve the existing problems with the fixed weight allocation scheme, we propose a dynamic weight allocation method. The simulation results show that it can improve the fair share allocation and the overall QoS of all the ABR connections. We also study the effect of RM cell frequency, control delay and stability factor.

The thesis is organized as follows. Chapter 1 introduces the thesis work and related research in the area. In Chapter 2, we review the traffic control mechanism and source rules of the ABR service as well as rt-video transporting techniques. In Chapter 3, we describe the rate control algorithm and its modeling using the OPNET tool. In Chapter 4, following extensive simulation campaigns, we examine the performance of the FASTRAC rate control algorithm and the effect of various parameters and factors. Based on our simulation results, we propose some solutions to improve the overall performance when supporting rt-video traffic. In Chapter 5, we summarize our conclusions and suggestions based on the simulation results and analysis.

CHAPTER 2 ABR TRAFFIC CONTROL AND rt-VIDEO COMMUNICATIONS

2.1 ABR Traffic Control

2.1.1 Overview of ABR Service and ATM QoS

The ATM-Forum Traffic Management Specification (TM4.1) defines the following six service categories for ATM networks [1].

CBR - Constant Bit Rate

rt-VBR - Real-Time Variable Bit Rate

nrt-VBR - Non-Real-Time Variable Bit Rate

UBR - Unspecified Bit Rate

ABR - Available Bit Rate

GFR - Guaranteed Frame Rate

Each service category is defined using a traffic contract and a set of QoS parameters. The traffic contract is a set of parameters that specify the characteristics of the source traffic. The ATM Forum specifies four traffic parameters – Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), Maximum Burst Size (MBS), and Minimum Cell Rate (MCR). A tolerance value for PCR, called Cell Delay Variation Tolerance (CDVT), is also specified. This defines the allowable variation from PCR for a connection's traffic to be conforming. Three QoS parameters are defined by the ATM Forum: two delay performance parameters (peak-to-peak CDV and maxCTD) and one dependability parameter (CLR). These QoS parameters are negotiated by the source with the network, and are used to define the expected quality of service provided by the network. For each service category, the network guarantees the negotiated QoS parameters if the end system complies with the negotiated traffic contract. For non-compliant traffic, the network does not need to maintain the QoS objective.

ABR is an ATM layer service category for which the limiting ATM layer transfer characteristics provided by the network may change subsequent to connection establishment. On the establishment of an ABR connection, the end-system shall specify to the network both a maximum required bandwidth and a minimum usable bandwidth. These shall be designated as peak cell rate (PCR), and the minimum cell rate (MCR), respectively. The MCR may be specified as zero. The bandwidth available from the network may vary, but shall not become less than MCR. The bandwidth allocated by the network to the ABR connection may vary between MCR and PCR according to the changing conditions of the network. The network tries to maintain a low cell loss ratio and loose cell delay by providing the sources with feedback information regarding the changing of the available bandwidth. This feedback is conveyed to the source through specific control cells called Resource Management cells, or RM-cells. It is expected that an end-system that adapts its traffic in accordance with the feedback will experience a low cell loss ratio and obtain a fair share of the available bandwidth according to a network specific allocation policy.

The ABR service does not limit or guarantee the delay or the delay variation experienced by a given connection. In other words, the ABR service was not designed to support real-time applications. However, PCR and MCR traffic parameters have been specified for the ABR service. Once the ABR connection is established, the bandwidth available from network may vary, but shall not become less than MCR. So the ABR service can guarantee to provide a minimum QoS as specified by MCR specified. Moreover, ABR may be able to provide additional bandwidth to MCR, depending on the available bandwidth of the network and allocation policy. If the end-system applications can adapt their traffic properly according to the feedback of the network bandwidth availability, the Cell Loss Ratio (CLR) should be very small and the delay variation should be bounded as well. It's possible to support rt-video applications over ABR service with loose QoS. It depends on how fairly the switches provide the sources with network available bandwidth information and how accurately and responsively the sources (applications) adapt to the changing condition of the network.

2.1.2 ABR Flow Control Mechanism

ABR flow control occurs between a sending end-system (source) and a receiving end-system (destination). A source generates forward RM-cells, which are turned around by the destination and sent back to the source as backward RM-cells. These backward RM-cells carry feedback information provided by the network switches and/or the destination back to the source. A network switch may:

- Directly insert feedback control information into RM-cells when they pass in the forward or backward direction.
- Indirectly inform the source about network congestion by setting the EFCI bit in the data cell header of the cells of the forward information flow. In this case, the destination will update the backward RM-cells based on this congestion information.
- Generate backward RM-cells.

The ABR traffic management model is a rate-based end-to-end closed-loop feedback control model. The feedback mechanisms have been typically either binary or explicit rate, i.e., a specific rate is sent to the source through the RM-cell. There are three ways for ABR switches to provide feedback information to the sources:

(1) Explicit Forward Congestion Indication (EFCI) bit, contained in ATM header, can be set by the switch to indicate congestion state of the congested switch. Such switches are called Binary or EFCI switches.

(2) The congested switches can set the Congestion indication (CI) bit and the No Increase (NI) bit in the payload of the Resource Management (RM) cell. The sources can use this feedback information to increase, decrease, or keep its rates by pre-defined factors, but there is no explicit rate provided for the

sources. The switches that only use this mechanism are called Relative Rate Marking.

(3) Switches indicate to the sources the rate at which they can send the cells into the network. This rate is conveyed in the Explicit Rate (ER) field of every RM cell. Such switches are called Explicit-Rate switches. The ER switches compute the Explicit Rate for each of the sources according to several parameters, such as source requested rate, MCR and/or PCR, and fairness scheme or policy.

The ATM Forum specifies the switch behavior to implement at least one of the three aforementioned methods to control congestion at queuing points. A switch may also generate a backward RM-cell (BECN) in case of being extremely congested. But this out-of-rate RM cells are limited to a rate specified by parameter TCR (Tagged Cell Rate) that is fixed to 10 cells per second per VC [1]. The backward notification method can reduce response time, but the nodes (switches and destination end-system) will not have a coordinated view of the correct rate. The ATM Forum standards specify the feedback information that the switches should provide to the sources as well as the formats of the RM cells and parameters. There is no specification on how the switches compute the current fair shares of available bandwidth (or explicit rate) for each connection or how the switches determine when to activate EFCI (or CI/NI) for each connection. A number of ABR rate-based control algorithms have been proposed and simulated over the years. Some of them have been listed in the ATM Forum TM 4.1 specifications, such as the

Enhanced Proportional Rate Control Algorithm (EPRCA) and the Explicit rate Indication for Congestion Avoidance (ERICA). The most crucial performance issue is left open for further studies. Current studies of ABR switch mechanisms are focused on how to achieve fairness, responsiveness, stability and simplicity.

Fairness is obviously the most important criteria of the switch ABR control algorithm. However the definition of fairness is not unique. There have been several different fairness criteria proposed in the literature. They all have different definitions for fairness. At an intuitive level, a bandwidth allocation scheme is fair if it does not offer a different treatment to connections, either based on the time order in which they request a share of the available bandwidth, or on the particular location of their source and destination end points [24]. The ATM Forum TM4.1 lists six example fairness criteria which are: Max-Min, MCR plus Equal Share, maximum of MCR or Max-Min Share, Allocation proportional to MCR, Weighted Allocation and MCR plus Weighted Share (General Weighted Fairness) [1] [13]. Some of them are special cases of others. For instance, Max-Min, MCR plus Equal Share and Allocation Proportional to MCR are all special cases of the General Weighted Fairness. A reasonable amount of flexibility in the interpretation of fairness should be left to ABR service providers. The issue of fairness is intimately related to the way network service providers manage their networks in terms of service billing and other complex network optimization problems [24].

The fair distribution of bandwidth to different connections is an important issue in the Available Bit Rate (ABR) service [25]. Traditional schemes are generally based on connection weight that can't reflect the dynamic changing demand of the ABR sources. They fail to achieve desired bandwidth allocations and link utilization when supporting rt-video applications. In this thesis, a new scheme called dynamic fair share scheme is proposed which substantially improves fairness in bandwidth allocations. Details of the scheme are discussed in next chapter.

2.1.3 Rules of ABR Source

The ATM Forum TM 4.1 Specifications [1] specifies the rules that the sources have to follow. Here we only discuss some of them. The sources have three kinds of cells to send: data cell, FRM cells and BRM cells (corresponding to the reverse flow as a destination). The frequency of FRM cells are limited by three parameters: N_{rm}, T_{rm} and M_{rm}. The proper tuning of these three parameters can avoid the problem of generating too many or too few FRM cells. In practice, the choice of N_{rm} and T_{rm} affects the responsiveness of the control and computational overhead at the end systems and switches. The inter-RM-cell interval determines the responsiveness of the system. With larger inter-RM-cell interval the system becomes less responsive, but it takes less computational and bandwidth overheads. The ATM TM specification sets the upper bound for T_{rm} to 100 ms.

The source should react to network feedback that is specified using the ER, CI and NI bits embedded within the cell header. Normally a source could simply change its ACR to the new ER value; but this is not true in the following two cases. (1) If the new ER value is very high compared to the current ACR, switching to the new ER will cause sudden queues in the switch. So the ACR increase is limited by a factor RIF. The source cannot increase its ACR by more than RIF×PCR each time. (2) If there are EFCI switches in the connection path, they may set the EFCI bit to indicate congestion. In this case, the source ACR should be decreased multiplicatively [26].

The ATM Forum has detailed the specifications on the source behavior, especially on the way the source rate is changed in response to the feedback provided by the ATM switches. But this is only from the network point of view. The crucial problem is how the applications or higher layers adapt or react to the allowed source rate according to the feedback information from network. The study of this issue is currently of great interest in the networking community.

2.2 Real-time Video Transmission Techniques

Digital video contains a great amount of information and is among the most demanding applications in terms of storage space and transmission bandwidth. Subsequent pictures are usually similar and contain spatial redundancy within each frame and temporal redundancy between consequent frames. The MPEG video compression techniques take the advantage of these redundancies as well as human perception to reduce the bandwidth and storage requirements when transporting video application. A scalable encoding scheme allows the representation of a multimedia at different levels of resolution. The encoder generates data at different rate, allowing it adapt to the available network resource. This can be obtained by changing one or more of the following: refresh rate (frame rate), quantization and movement detective threshold. Another method is using hierarchical encoding as MPEG-2 scalable mode layered encoding. Video rate smoothing techniques can reduce rate variability of variable bit rate video data. These video processing techniques can be selectively used for transporting video applications over bandwidth limited services like ABR.

2.2.1 MPEG-2 Video Compression Scheme

The MPEG specifications include standards for compressing, multiplexing and reconstruction of video streams and consist of three parts: video, audio and system information. The video and audio information deal with encoding processes and the system information containing time stamping information that synchronizes the video and audio signal. The MPEG standards do not specify any particular encoding algorithm but they indicate the format (syntax) in which data should be presented to a decoder and also the rules of decoding process (semantics). Therefore, an encoder can use any encoding process to produce a bit-stream that compiles with the syntax and can be decoded according to the decoding semantics.

The MPEG compression algorithms use loose compression techniques. They take advantage of the redundancy characteristics of video, i.e., subsequent pictures are usually similar. An MPEG video stream contains three types of picture frame:

- I-frame (Intra-frame): Self-contained pictures that are coded as still images. These frames are encoded without reference to any other frames and thus can be used as decoding starting points. These frames are normally the largest among the three types because they can only be spatially compressed.
- P-frame (Predictive frame): Pictures that are coded with reference to a previous I-frame or P-frame and can be used as a reference point for picture P-frames or B-frames. They are usually smaller than I-frames because of temporal reduction.
- B-frame (Bi-directional frame): Pictures that are coded with reference to the closest two I-frames or P-frames, one encoded in the past and one encoded in the future. It can not be used as a reference point for other pictures. B-frames have the highest level of compression.

MPEG video bit-streams are organized in groups of pictures. A Group Of Pictures (GOP) normally contains one I-frame, a few P-frames and possibly some B-frames. An MPEG sequence may look like: IBBPBBPBBPBBIBBP...... The structure of a GOP is specified by two parameters: the number of pictures in it (*N*) and the number of B-frames between two successive P-frames (*M*). Obviously, such a sequence of data results in a very bursty traffic due to the difference between picture sizes. Also, since changes cause variation of size within a certain type of frames. Such traffic is called Variable Bit Rate

(VBR). Because bursty traffic have undesirable impacts on networks, MPEG encoders can also produce Constant Bit Rate (CBR) video streams where frame sizes do not vary a lot but the picture quality could be poor. The reason for this is that, in order to maintain a fixed bit rate, pictures with different sizes need to be encoded at different Quantization Parameters (Q-factors).

Multimedia compression is achieved by reducing the redundancy existing in digital audio, image and video (both spatial and temporal) and taking advantage of the properties of human perception. There are several main compression techniques used for video: discrete cosine transformation (DCT) based, wavelet transform based, proprietary compression methods, vector quantization and content-based method [27].

DCT is a compression method used in the popular MPEG-2 standard. The MPEG-2 compression methods can be used to support rate adaptation in the following aspects. First, the rate of the source can be changed using different quantization levels and encoding rates. Second, scalable video is permitted in MPEG-2 video; four scalable modes encode the MPEG-2 video stream into different layers (base, middle, and high layers), mostly for prioritizing video data. There are two main purposes. First, important video data are placed in high priority to protect error. In case of network congestion, less important, low-priority data are dropped first and high-priority data are kept so that a reasonable video quality can still be reconstructed from these data. This is so-called graceful QoS degradation. Second, scalable video allows the decoder to selectively decode part of a video (e.g. a standard resolution TV receive and decode so coded HDTV program) [28].

In Wavelet compression, the coder performs a set of filtering operation on an image to divide it into spectral components with increasing resolution. Each component represents a sub-image. Image data in each sub-band is transformed using a wavelet function to obtain transformed coefficients. The transformed coefficients are then quantitized and run-length encoded before transmission. In a sense, wavelet compression results in progressively encoded video. Two common approaches for wavelet compression are to use a motion-compensated 2-dimensional (2-D) wavelet or a 3-D wavelet function [29] [30]. Wavelet compression overcomes the blocking effects of DCT based methods since the entire image is used in encoding instead of blocks. An important feature of wavelet transforms is the support of scalability for image and video compression. Wavelet transforms coupled with encoding techniques provide support for continuous rate scalability, where the video can be encoded at any desired rate within the scalable range.

2.2.2 Scalable Encoding Techniques

The multimedia application can adapt to available bandwidth change by changing dynamically the compression parameters, e.g. quantization level. This requires a flexible coding scheme, one that operates at a range of compression ratios, and a scheme to compute the available bandwidth for the multimedia stream (e.g. explicit feedback scheme of ABR) and to estimate the quality of reception of the receiver.

A scalable encoding scheme allows the representation of a multimedia at different levels of resolution thus the encoder output different data rate that adapts to the available network resource. This can be obtained by changing one or more of the following: refresh rate (frame rate), quantizer and movement detective threshold. Another method is using hierarchical encoding as MPEG-2 scalable mode layered encoding [15] [31].

With a scalable layered encoding, a digital MPEG-2 video stream is encoded into two sub-bitstreams. This partitioning is done by taking into account the relevance of the different pieces of information of the MPEG-2 encoded bitstream. Under this scheme, low frequency coefficients together with other relevant information are transmitted at a high priority level, also denominated base layer. High frequency coefficients and other less important information are then transmitted at a lower priority level, also denominated enhancement layer. The breakpoint marks the division between the base and enhancement layers. If parts of the enhancement layer are lost, they are simply replaced by zeros and the image is reconstructed using the base layer and the dummy enhancement layer. The resulting image -though somewhat distorted- may be acceptable. Figure 1 depicts the bit-rate distribution of the clips and corresponding base layer. We can observe the amount of traffic reduction obtained from the partitioning.

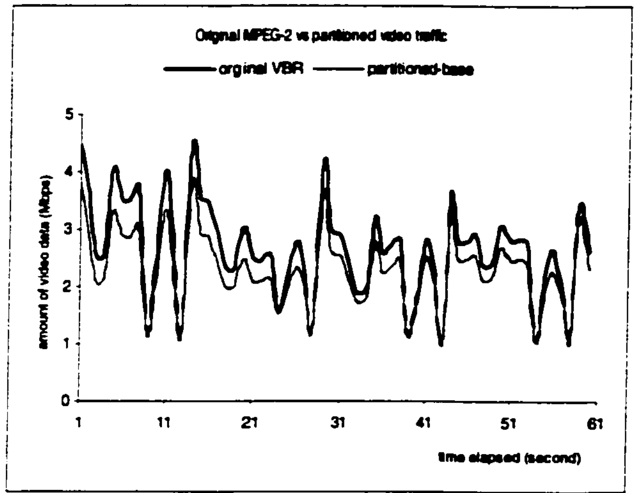


Figure 1 Original MPEG-2 video vs. partitioned base layer video traffic

This scheme can be easily adapted to transmission networks that support different priority levels, such as the ones used in ATM-based networks. By using this scheme a correct transmission of the most important information of the video signal can be somehow guaranteed. Furthermore, the base layer can be designed so that it can provide by itself a minimum acceptable image quality in situations in which the enhancement layer is completely lost.

2.2.3 Smoothing Techniques

One way to reduce the rate variation of the multimedia application is to perform smoothing of the transmitting stream. For live (non-interactive) video, a sliding window of buffers can be used at the source to smooth the variation of the video stream rate. The smoothing buffer can be drained at a rate adapting to the network available bandwidth.

The encoder dynamically changes its quantizer parameter according to the feedback about the network as well as the smoothing buffer [16] [17]. For pre-recorded (stored) video, a priori knowledge of the number in each frame (frame size) of a video sequence can be utilized to smooth the video traffic. Bandwidth smoothing techniques can reduce the burstiness of a variable-bit-rate stream by transmitting data at a series of fixed rates.

A smoothing operation helps improve the ability of the network to provide deterministic quality of service guarantees to a flow. Indeed, the smoothed video traffic is easier to characterize: the VBR traffic pattern is translated into a sequence of consecutive CBR transmission intervals. Also, the network resources required for the transmission of the video stream are reduced as a result of the smoothing.

The smoothing process generates a transmission schedule consisting on consecutive intervals where the transmission rate is constant and where a constant bandwidth can be reserved from the network. Among the different types of smoothing algorithms evaluated in [32], we chose to describe in detail the optimal smoothing scheme proposed in [33]. This scheme is according to the classification made in [32] a Minimum Variability Bandwidth Allocation type of smoothing algorithm. The algorithm is optimal in the sense that it minimizes both the transmission peak rate and the sample variance of the transmission rates.

The notations used in the formulation of the algorithm are as follows. N denotes the number of frames in the original video sequence, $t \in \{1, \ldots, N\}$ is the frame index (i.e. time index). The introduction of a start-up delay, s, extends the video sequence length by s frames. We denote by $d(t)$, $t \in \{s+1, \ldots, N+s\}$, the amount of data consumed by the client at time t (i.e., the size of t^{th} frame); we have for $t \in \{1, \ldots, s\}$, $d(t) = 0$. Let $a(t)$, $t \in \{1, \ldots, N+s\}$, be the amount of data sent by the server at time t. Over a time interval $[1, t]$, the cumulative amounts of data consumed by the client and sent by the server are given by: $D(t) = \sum_{i=1}^{t} d(i)$ and $A(t) = \sum_{i=1}^{t} a(i)$, for $t \in \{1, \ldots, N+s\}$. The cumulative amount of data that can be received by the client over a time interval $[1, t]$, is limited by the buffer capacity, b. Its maximum value is: $B(t) = \min\{D(t-1) + b, D(N+s)\}$, for $t \in \{2, \ldots, N+s\}$, with $B(1) = b$ and $B(0) = 0$. A feasible server transmission schedule is a $N+s$ – dimensional real vector

$$S = [a(1), \ldots, a(N+s)]$$

such that the client buffer never starves and never overflows (see Figure 2).

We should note that, given the transmission schedule obtained from the smoothing operation, the resource reservation does not have to follow exactly the source rate changes. Indeed, since the smoothing mechanism has reduced substantially the burstiness of the original VBR traffic, the provisioning of deterministic guarantees to the video flows can be met while maintaining high network utilization. For instance, a bandwidth reservation schedule can be constructed by considering local maximum transmission rates or even the peak transmission rates, since it has been minimized by the smoothing algorithm. Another advantage of these alternative approaches is the fact that they reduce the signalling traffic generated during the transmission of a flow; which can be significant in a network handling a large number of flows.

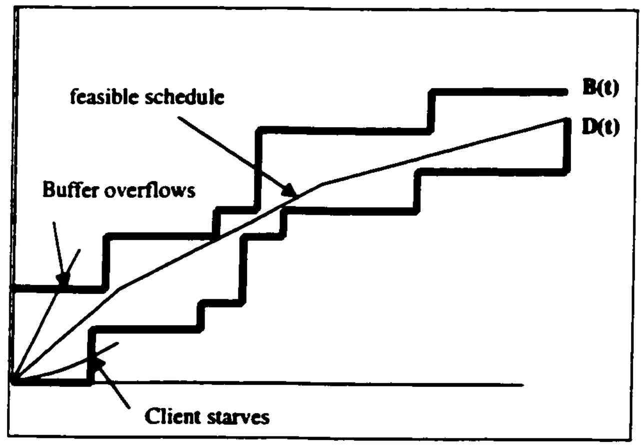


Figure 2 Transmission Schedule

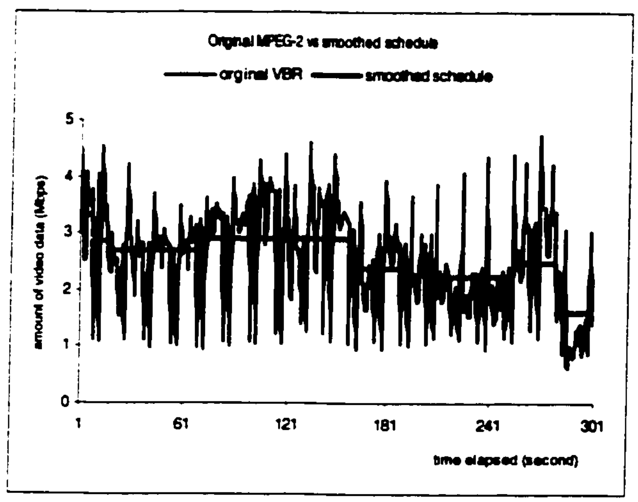


Figure 3 Original MPEG-2 video vs. smoothed transmission schedule

Figures 3 illustrates the reduction of burstiness obtained for a 5-second video sequence from the movie "Bear", smoothed using zero playback delay and 64Kbyte client buffer size.

[22] presents a comparison of six smoothing algorithms. The paper compares the transmission schedules generated by the various smoothing algorithms, based on a collection of metrics that relate directly to the server, network, and client resources necessary for the transmission, transport, and playback of prerecorded video. Each of the algorithms achieves different performance metrics although the primary goal of each of them is to reduce the peak transmission rate of the video. To bridge bandwidth smoothing and rate adaptation is an issue to be further studied.

CHAPTER 3 THE FASTRAC RATE CONTROL ALGORITHM AND MODELING

3.1 Introduction

The FAST RAte Computation (FASTRAC) rate control algorithm proposed in [23] is a simple, scalable, and provably stable fair rate allocation algorithm. The algorithm does not require the collection or analysis of state information on individual connections (e.g., per-connection rates, per-connection queue sizes, etc.) in order to provide stable and fair rate allocations to connections. Instead it only maintains the measurement of the aggregate background traffic rate and the total ABR traffic rate. The algorithm also takes into account the control loop delay. It is scalable to the network configuration and the number of connections.

The FASTRAC rate control algorithm has the following important attributes:

- The algorithm uses two primary measurements, the aggregate background (CBR/VBR) traffic rate and the aggregate ABR traffic rate. The algorithm does not require per-connection rate measurements or variables.
- The quantitative analysis has proven the stability and equilibrium performance of the algorithm.
- The explicit rate (i.e., the fair rate allocation) is computed only once every t_m seconds (typically in the range of 0.5 ~ 1 ms for an OC3 link). This has an

important impact on the processing overhead expected from the switch controller. This also implies that no dedicated hardware is required for implementing the rate computations. The computation can be done entirely in software. The only hardware assistance needed is for writing the fair rate computation into the RM cell, which has to be done at wire speed.

- The algorithm achieves *Max-Min* fairness and support MCR guarantees (*MCR* >0) in a natural way without any additional computation or information about bottleneck rates.

The fair rate allocation computed for an ABR source can be expressed by the following integral control equation:

$$r(n+d+1) = r(n+d) + \alpha \frac{T(n) - M(n)}{N} \qquad (1)$$

where,

$r(n)$ is the fair rate allocation computed for a source at time n. For a persistent source (i.e., a source having always data to transmit) that is also rate compliant, $r(n)$ is assumed to be the actual load arriving at the network node from the source.

$M(n)$ is the estimate of the aggregate traffic load of ABR connections on the network node derived from sample measurements.

t_m is the sampling interval, which is the time elapsed in the interval $[n, n+1]$, n being discrete time instants.

T is the target load available to ABR connections that we wish to reach at the network node. For a variable target load we have, $T(n)$.

α is a *control gain* and $[T(n) - M(n)]$ is an error (or rate mismatch) term between the target load available to ABR connections and the actual measured aggregated ABR traffic.

d is the time interval (in units of t_m) it takes for a control decision sent by the bottleneck switch to take effect at the bottleneck switch. If *RTT* (in units of t_m) is the round-trip time from the source to destination and back, then $d \le RTT$ depending on the location of the bottleneck switch from the source. We have $d = RTT$ as the extreme case if the destination is directly attached to the bottleneck switch.

N is the number of active ABR connections. We assume that the N connections share equally the effort in reducing the rate mismatch between T and M. This is the case when the *Max-Min* fairness scheme is used for fairness rate allocation among the ABR connections.

If each ABR source has different weight for share allocation, the fair rate allocation computed for an ABR source i can be expressed by the following integral control equation:

$$r_i(n+d+1) = r_i(n+d) + \alpha \frac{[T(n) - M(n)]W_i}{\sum_{j=1}^{N} W_j} \qquad (2)$$

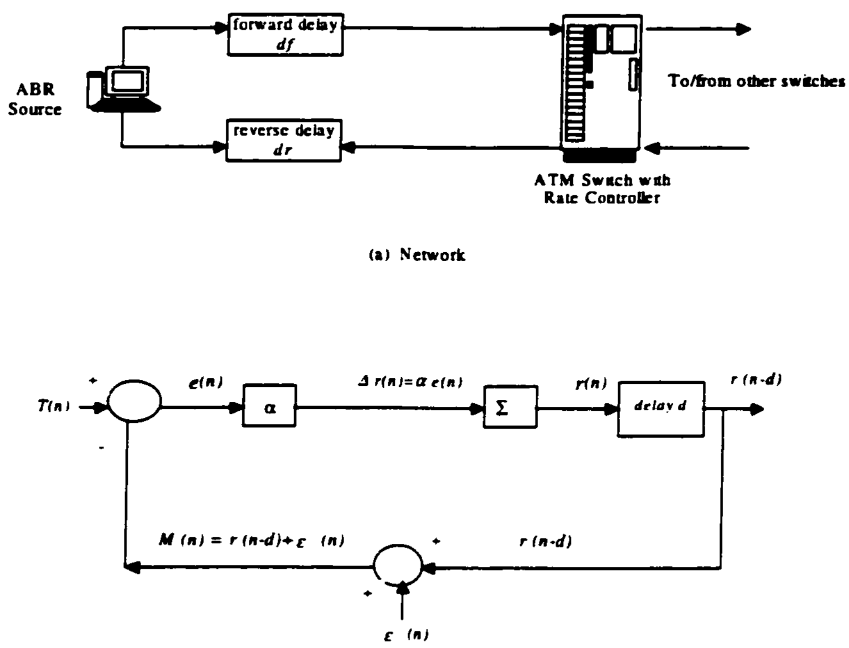
where:

W_i is the fair share weight of source i,

$\Sigma(W_j)$ is the total fair share weights of all active ABR sources.

It is worth to mention that in the ATM Traffic Management specification, the weights are fixed for the ABR connections. In chapter 4 we will introduce the use of a dynamic weight allocation scheme.

The explicit rate control mechanism is a closed-loop control model and can be illustrated in the following figures. The FASTRAC algorithm operates at each ATM switch that supports ABR connections. The switch monitors the total available ABR capacity (*T(n)*) and total ABT load (*M(n)*). The rate fair share for each ABR connection is computed every t_m second based on the following information: total target ABR capacity, measured total ABR load, fair share allocation criteria and ABR queue length.



(a) Network

(b) Feedback control scheme

Figure 4 Rate control scheme

The computed explicit rate at the bottlenecked switch is then sent to the ABR source via the ER field in the backward RM cell. The parameter d_r represents the time delay it takes to send rate allocation signals (explicit rate) from a switch at a bottleneck node to the data sources, and d_f represents the delay it takes for the rate allocations to become effective at the bottleneck node. The total control loop delay is the sum of the forward and reverse delay, and is equal to $d = d_r + d_f$. Figure 4 (b) illustrates the equivalent closed-loop control scheme.

The rate controller at the network node regulates the source's traffic rate to the network in proportion to the difference between the estimated traffic load and the desired load level. This is done using explicit rate control signals. Where we assume that the N sources share the effort in reducing the rate mismatch between T and M. The parameter N can be adapted dynamically to track the variations in the number of active connection if required. This helps to enhance the responsiveness of the algorithm, especially when the number of connections can be large but only a few are active at a given time [23].

The stability analysis has proven the following propositions:

Proposition 1: A necessary condition for the flow control process established by equations (1) and (2) to have asymptotic internal stability is

$$0 < \alpha < 1/(d+1) \qquad (3)$$

Proposition 2: For multiple sources with different feedback delays $d_1 < d_2 < \ldots d_i \ldots < d_N$. and with the possibility that some sources can go silent or inactive, a value of $\alpha < 1/(d_N + 1)$ is necessary to guarantee the stability of the entire flow control process.

The above proposition was verified extensively through simulation [23]. Thus, in our flow control algorithm, the value of α used for multiple sources with different feedback delays is

$$\alpha < \frac{1}{d + \frac{1}{2}} \qquad (4)$$

where $d = max\,(d_1\ , d_2\,, \ldots d_i \ldots\ d_N\,)$

The value of d is derived based on the round-trip time (*RTT*) of the flow control process which takes into account the propagation delay, queuing delay and processing delay. Thus, for N connections through a bottlenecked node, the delays are usually much less than the maximum round-trip delay *maxRTT*, making (4) a more conservative stability criterion. This means, apart from (4) being less than the upper bound (3), the condition (4) is based on the maximum *RTT* which ensures stable system performance for all conditions. It will be shown later that this stability condition (4) makes the rate control process behave like a recursive filter with memory equal to d time units.

The dynamics of the system is affected significantly by different configurations of α. When α is large (i.e., $\alpha \rightarrow 1/(d + 1/2)$), the system reaches steady state much faster but shows more oscillating transients. For a smaller α (i.e., $\alpha \to 0$), the system reaches steady state slowly and has smaller oscillating transients. Overall, the algorithm is configurable according to the needs of the system.

To take into account the effect of control delay to the bottleneck switch, the following formula is used to determine the value of the α.

$$\alpha = \frac{f}{1 + round[\frac{d}{\sigma}]} \qquad (5)$$

where:

f is the stability factor used to adjust the stability of the rate control,

d is the maximum among the control delays of ABR connections on the switch,

σ is the measurement update interval at the switch.

Based on the quantitative analysis above, the FASTRAC rate control algorithm is stable and can achieve fairness if condition $0 < \alpha < 1/(d+1/2)$ meets. The previous numerical simulations also show that it achieves stability and fairness when supporting greedy traffic, such as file transfer. The thesis extends the study by exploring the behavior and performance of the algorithm when it supports real-time video traffic. The thesis also studies the performance of the algorithm with different conditions and parameters. The section below will introduce the modeling of the algorithm using the OPNET simulation tool [34].

3.2 Simulation Modeling

- **Network Model**

To achieve generality of our study, the following different network configurations are used for our simulations: the bottleneck configuration and the Generic Fairness Configuration (GFC) with bottleneck link at different switches. As shown below in Figure 5, 6, and 7 respectively, each configuration contains explicit rate control switches using the FASTRAC rate control algorithm and ABR sources that are capable to adapt to the feedback rates from the network. The VBR source generates self-similar traffic to simulate the aggregated background VBR traffic on the bottlenecked link. We assume the bottleneck link of the network is always the same but it is not always under overload condition. The VBR connection has higher priority over the ABR connections. The ABR sources utilize the unused bandwidth left by the VBR connection and try to adapt to the changing of the available bandwidth on the network.

(1) Bottleneck Configuration

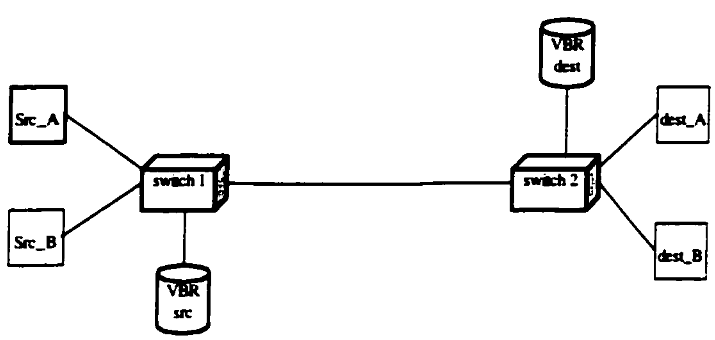


Figure 5 Bottleneck configuration

(2) Generic configuration with bottleneck link at switch 3

As shown in Figure 6 below, the generic configuration consists of 5 ABR switches, 6 ABR nodes (source and destination) and 1 VBR node (source and destination). The bottleneck link is at switch 3. There are six ABR connections and the control delays of the connections are not the same.

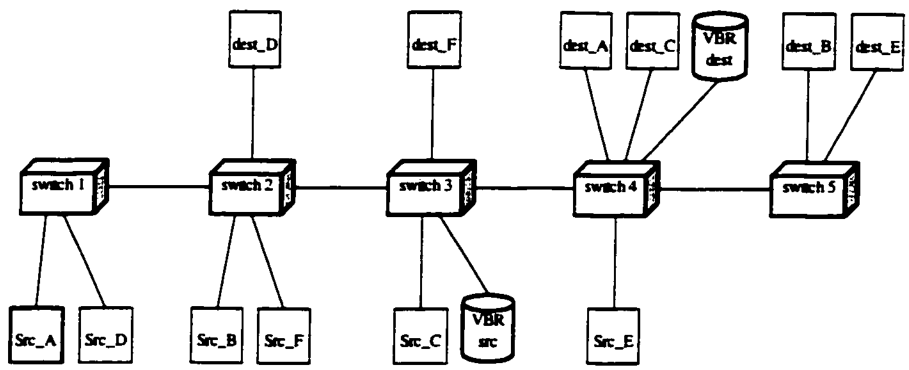


Figure 6 Generic Fairness Configuration with bottleneck link at switch 3

(3) Generic configuration with bottleneck link at switch 2

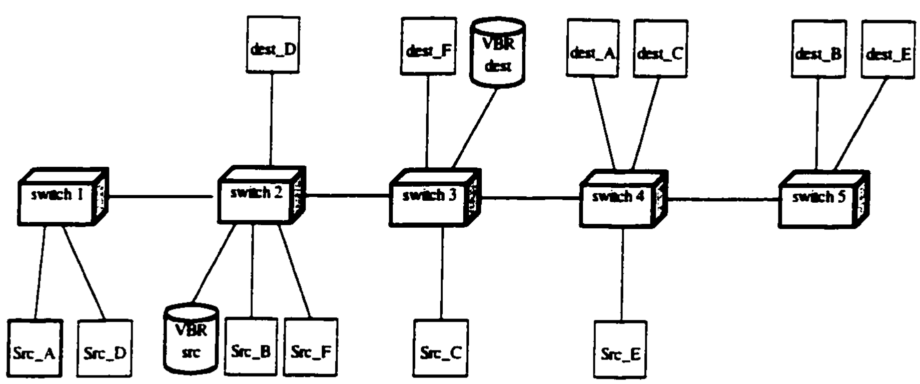


Figure 7 Generic Fairness Configuration with bottleneck link at switch 2

The Generic Fairness Configuration with the bottleneck link at switch 2, as shown in Figure 7 above, makes it an example network configuration with shorter control delay. Other configurations are the same as Figure 6.

- **Switches**

ABR explicit rate control scheme: the FASTRAC rate control

The capacity of each link is selected in so that the bottleneck link is at the switch at which the VBR source generates background traffic. All the ABR real-time video sources are competing the limited available bandwidth on the bottleneck switch. We consider the following link capacity as an example:

Generic configuration: 10Mbps (switch 1), 30Mbps (switch 2), 30Mbps (switch 3) and 20Mbps (switch 4)

Bottleneck configuration: 20 Mbps

Target Link Utilization: 95%

Fairness allocation criteria: fixed weight (*Max-Min*) rate allocation scheme and the proposed dynamic weight allocation scheme that is described in the following chapter.

- **ABR Sources and Background VBR**

(1) VBR Source (background traffic): self-similar traffic

(2) ABR Sources:

Each ABR source can transmit either greedy traffic or rt-video traffic, depending on the configuration. The ABR source supporting greedy traffic always has traffic to send and is dynamically adapting its transmission rate to the allowed cell rate (ACR) at any time. For ABR source that supports rt-video traffic, we assume the ABR source is capable of adapting to the allowed cell rate (ACR) by dropping the data of a frame that exceeds the ACR. We assume the ABR source supporting rt-video application does not generate more data when it has more fair share than it needs. The ATM defined parameters are set to their default values otherwise specified in the simulations below.

- **Fairness Allocation Criteria:**

There are several fairness allocation schemes suggested in the ATM Forum Traffic Management Specifications. In this thesis we use the *Max-Min* fairness rate allocation scheme as the fixed weight rate allocation scheme. To improve the rate allocation and overall QoS of all connections, we introduce a dynamic weight fairness allocation scheme. The proposed dynamic weight allocation scheme is discussed in the following chapter.

Other parameters such as link delay, fairness allocation scheme and stability factor are specified in each simulation.

CHAPTER 4 SIMULATION RESULTS AND ANALYSIS

Previous studies have proven that the FASTRAC rate control algorithm achieves fairness allocation over ABR sources characterized by greedy traffic. The nature of real-time video traffic is different to that of greedy traffic. With greedy traffic, the ABR source can always transmit at the rate specified by the fair share allocated to it according to the changing of the available network bandwidth for ABR connections. Therefore the link utilization of the bottlenecked link can always achieve the target with very little oscillation. The ABR source supporting greedy traffic can be considered as an ideal self-adaptive end-system. Due to the nature of real-time video traffic, the ABR source can not always adapt to its fair share allocated by the network. The simulation experiments in this thesis evaluate the performance of the FASTRAC rate control algorithm when supporting real-time video traffic over ABR service. We also study the effect of some parameters on the performance of the ABR system. The simulation experiments study the following issues:

- The effect of the various parameters and conditions, such as RM cell interval, stability factor and control delay, on the performance and responsiveness of ABR system using FASTRAC rate control scheme;

- The performance and behavior of the FASTRAC rate control algorithm when supporting real-time video traffic over ABR service in different network configuration scenarios;
- How the proposed dynamic weight fairness allocation scheme improves system performance compared to the fixed weight allocation scheme when transporting rt-video over ABR service;
- How the non real-time applications (e.g. file transfer) can take advantage of the unused fair share otherwise would be wasted by the real-time video ABR connections.

4.1 Effect of Parameter and Factor

4.1.1 Effect of RM cell interval

As described in the previous chapter, ABR traffic management mechanisms allow network to calculate the available bandwidth using its current load information. This in turn enables the allocation of the available bandwidth fairly and efficiently among the ABR connections according to the fairness criteria being considered. The rate feedback information is sent back to ABR sources via the Resource Management (RM) cells in the form of EFCI, NI, or ER. In the ABR traffic management framework, the ABR sources limit their data transmission to the rates allowed by the network. In this way, the more frequent the RM cell, the more instant and accurate the explicit rate. But the frequent transmission of RM cells results on the increase of management overhead of transmission. Most RM cells generated by the sources are counted as part of the network load in the sense that the total rate of data and RM cells should not exceed the ACR of the source. Our simulation experiments also show that a small RM interval does not always improve the responsiveness.

The figures below show the simulation results of fair share and ABR queue length. We use a generic configuration as shown in Figure 6 with the following attributes:

- measurement update interval at the bottleneck switch: *5 ms*.
- fair share allocation scheme: *Max.-Min*. (fixed weight of 1 for all ABR connections)
- link delay: *10 ms*
- greedy traffic for all ABR sources
- constant and unique stability factor, *f*, for all simulations

We use VBR background traffic with only a few rate changes (as shown in Figure 8) in order to see how RM cell interval is affecting the responsiveness. We vary the RM cell interval (N_{rm}) and examine the responsiveness of the allowed cell rate (ACR) and the ABR queue length at the bottlenecked switch. According to ATM specification, the N_{rm} must be a value from the following set [2, 4, 8, 16, 32, 64, 128, 256]. However, we only show the simulation results for N_{rm} = 8, 16, 32, 64. The N_{rm} values smaller than 8 incur a very high management overhead and are not realistic in real applications.

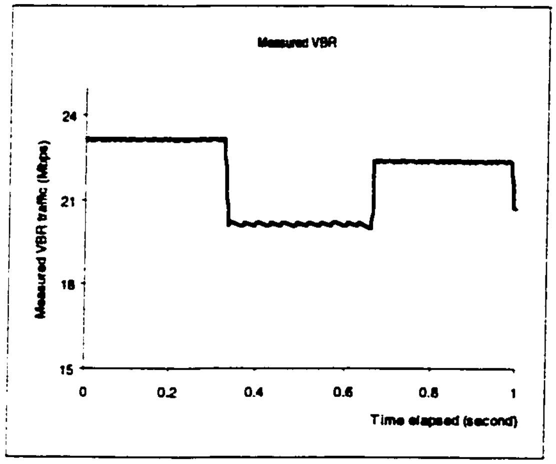


Figure 8 Background VBR traffic

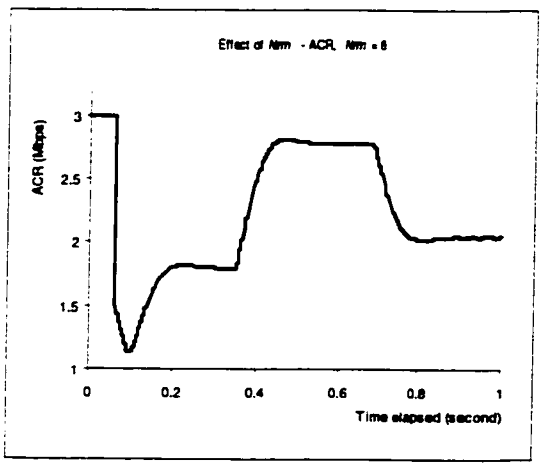


Figure 9 ACR of Src_A with $N_{rm} = 8$

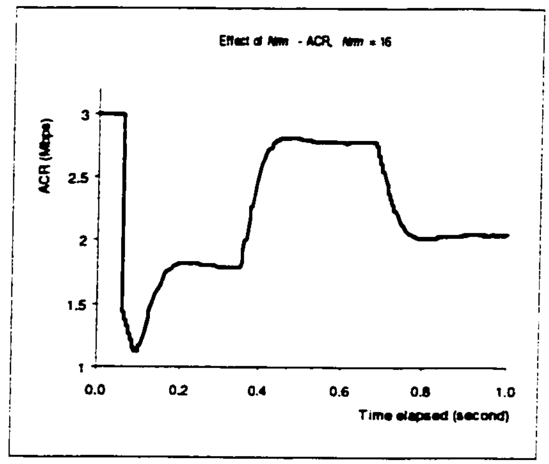


Figure 10 ACR of Src_A with $N_{rm} = 16$

From the results in Figure 9 to Figure 17, we can clearly see that with a small N_{rm} value (more frequent RM cells), the system has better responsiveness and smaller ABR queue size as the available ABR bandwidth changes. We also observe that the effect of different N_{rm} value is not obvious when they are small (N_{rm} =8, 16). This is because the RM cells become more frequent than the measurement update rate of the network. The update interval of the switch load measurement, $T(n)$ and $M(n)$, is set to 5 ms in this simulation. The RM cell interval is approximately 2 ms with N_{rm} of 8 at the rate shown above (1.7 Mbps) and approximately 4 ms for N_{rm} of 16. This means some RM cells will carry fair share information that has not yet been updated since the last BRM cell. However it is worth mentioning that under some conditions when ABR RIF source rule applies, increasing of RM cells can always improve responsiveness.

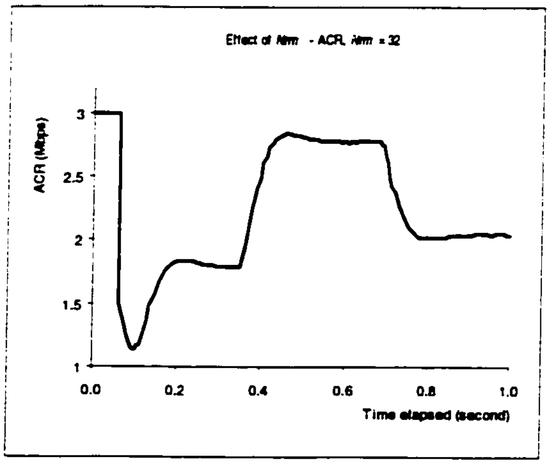


Figure 11 ACR of Src_A with $N_{rm} = 32$

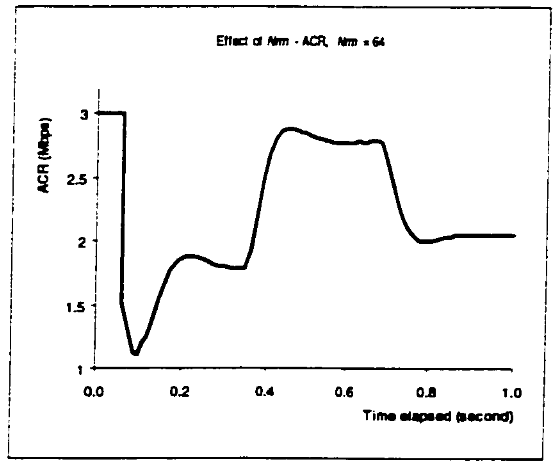


Figure 12 ACR of Src_A with $N_{rm} = 64$

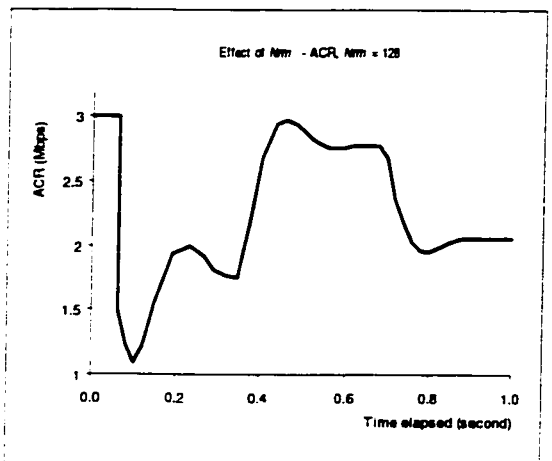


Figure 13 ACR of Src_A with $N_{rm} = 128$

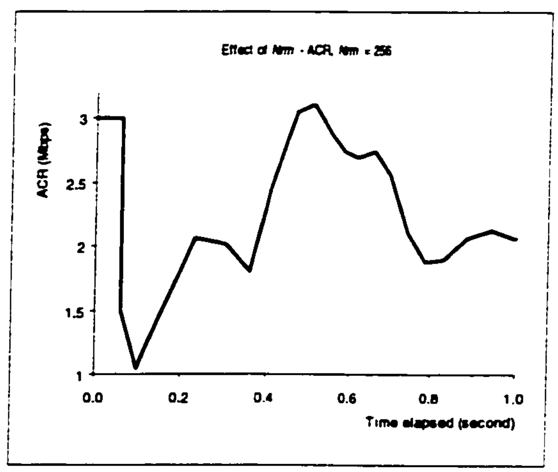


Figure 14 ACR of Src_A with $N_{rm} = 256$

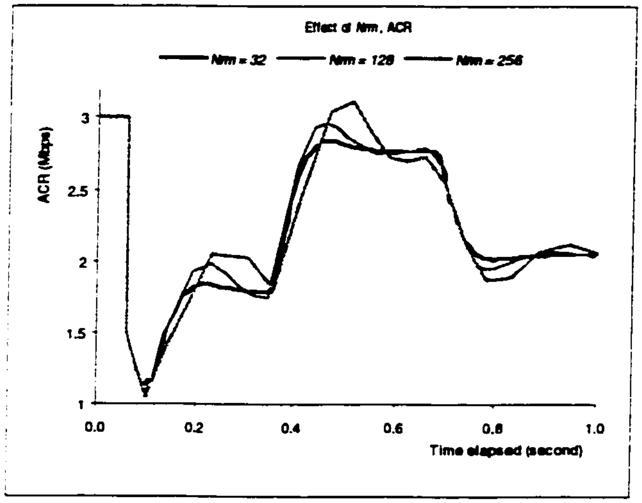


Figure 15 Comparison of ACR with different N_{rm}

Figure 15 illustrates the difference of responsiveness among three different N_{rm} values. When N_{rm} gets larger, the responsiveness gets worse, with the maximum N_{rm} (256) having the worst performance. Figure 16 and Figure 17 show the ABR queue size with N_{rm} value of 8 and 128 respectively.

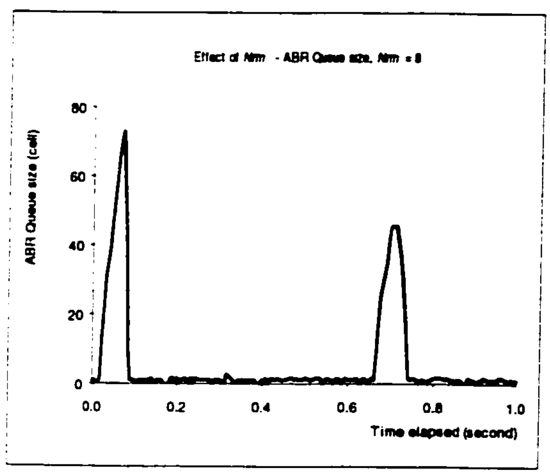


Figure 16 ABR queue size with $N_{rm} = 8$

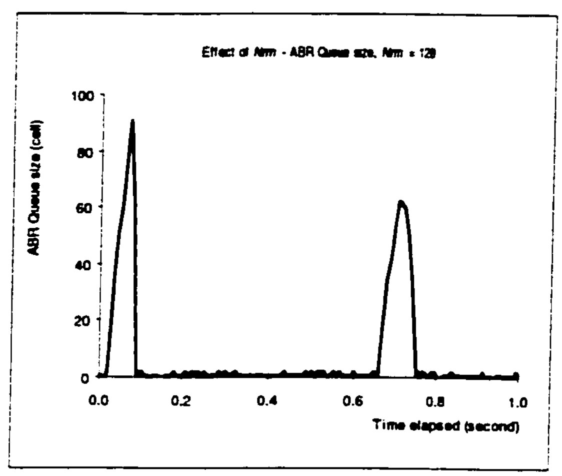


Figure 17 ABR queue size with $N_{rm} = 128$

Other factors such as sampling interval of load measurements at the switches and transmitting rate of ABR source also play an important role when selecting a good N_{rm} value. With the same switch measurement update interval (5 ms), Figure 18 below shows the result of N_{rm} = 128 with higher ABR transmitting rate (3.4Mbps). We can see that the same N_{rm} is good for an ABR connection with a higher rate (as shown in Figure 18) but is not good when the ABR rate is low (as shown in Figure 13). The selection of RM cell interval should consider both the responsiveness and the management overhead. When the time between RM cells is much smaller than the measurement update interval, there is very little improvement to the system responsiveness. A smaller N_{rm} value can still achieve good responsiveness when the ABR rate is high.

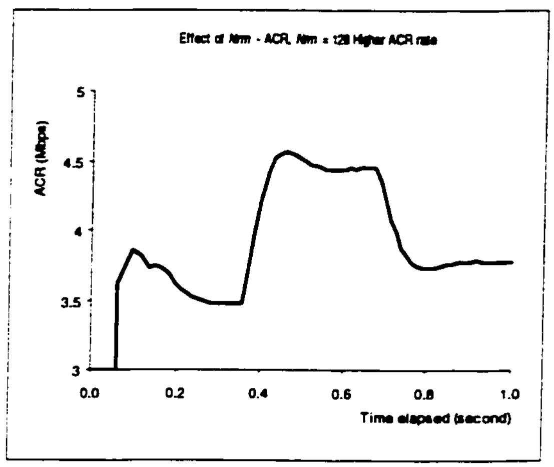


Figure 18 ACR of Src_A with same N_{rm} (128) but higher rate

4.1.2 Effect of Stability Factor

As shown in formula (1) and (2), the responsiveness of the system depends greatly on the control gain α. While α is restricted by (5) in order to guarantee the stability of the entire flow control process. While selecting a conservative value $\alpha = 1/(d+1)$ can ensure stability for all conditions, the tuning of α can significantly improve the responsiveness of the system in practice. The simulation results below show the effect of α (by tuning stability factor f) under different scenarios.

Figure 19 to Figure 20 illustrate the explicit rate for ABR Src_A in configuration shown in Figure 6 (Generic configuration with bottleneck link at switch 3). The configuration attributes are the same as in section 4.1.1 except here we vary the stability factor f instead of the RM cell interval. As depicted in (5), when changing stability factor f, the control gain α changes accordingly. To observe the effect, we use the same VBR background traffic in Figure 8 for this simulation.

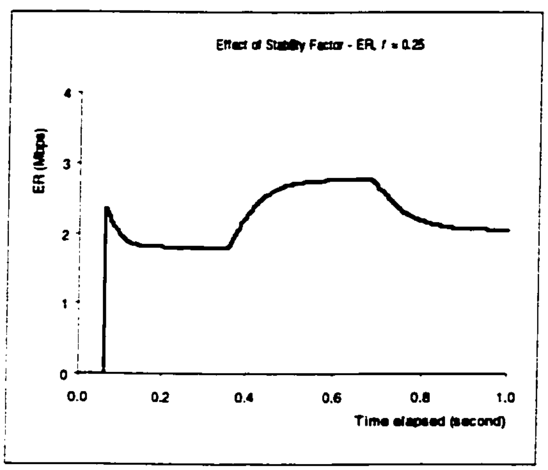


Figure 19 Effect of stability factor with f=0.25

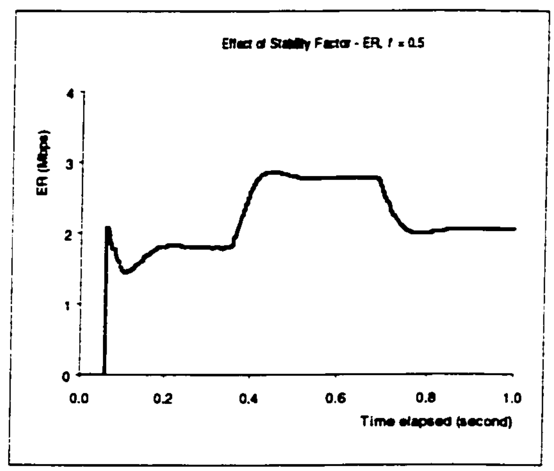


Figure 20 Effect of stability factor with f=0.5

From Figure 19 to 22, the results show that when VBR traffic changes from 23Mbps to 20Mbps, the explicit rate for ABR Src_A reaches its new steady rate gradually – no oscillation but takes long to converge to the new rate, the explicit rate converges to its new target faster but with more oscillations. But a bigger stability factor does not always make the system reach steady state faster. When the stability factor is too big, the rate oscillation is so large that it hardly reaches steady before the condition changes again.

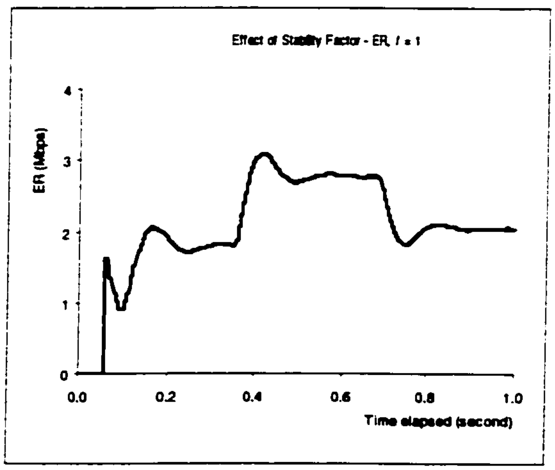


Figure 21 Effect of stability factor with f=1

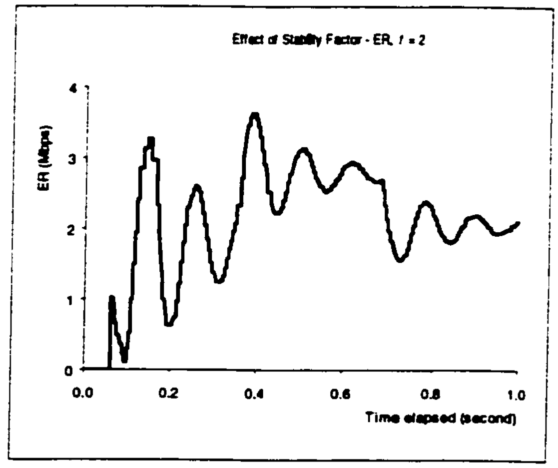


Figure 22 Effect of stability factor with f=2

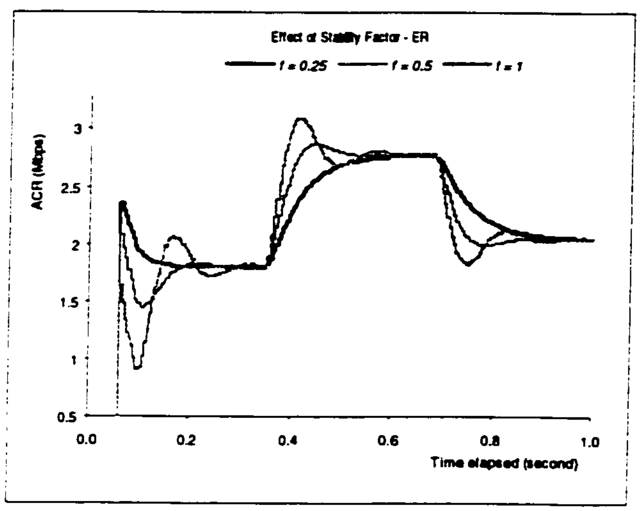


Figure 23 Comparison of ER with different stability factor values

can be explained by the nature of the FASTRAC rate control algorithm. With a smaller stability factor (f) hence a smaller control gain (α), the feedback explicit rates for ABR sources are gradually adjusting the mismatch (error) between the target and measured ABR load, one small fraction at a step. This conservative adjustment will take more steps to reach its new state but in a peaceful way. While with a larger stability factor (f), the rate adjustment step is bigger and it can result on over-adjustment. When over-adjustment occurs, re-adjustment in reverse direction is needed in order to reduce the mismatch. In such a violent way, computed explicit rate will oscillate. Figure 23 shows the difference of responsiveness among three different stability factor values. It clearly demonstrates the significance of the stability factor on the responsiveness.

The same behavior is also observed in the simulation experiments on the Generic configuration with bottleneck link at switch 2 (as shown in Figure 7). Figure 24 and Figure 25 show the results with different stability factors ($f = 0.25$ and 0.5).

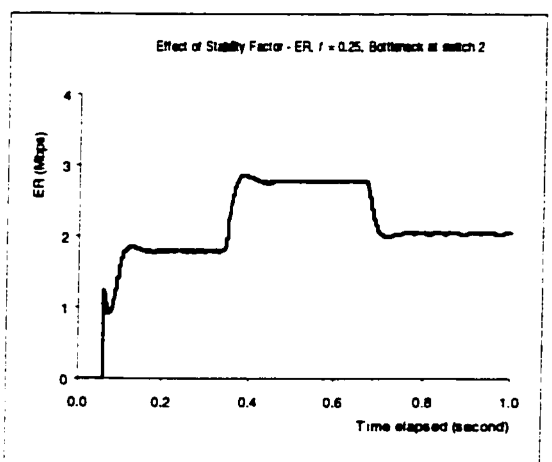


Figure 24 Effect of stability factor, f=0.25, at switch 2

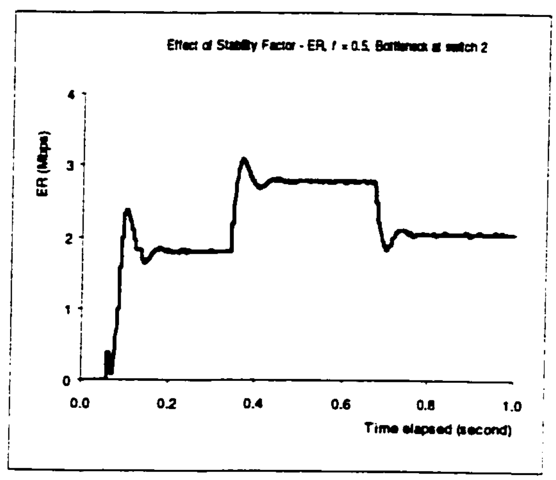


Figure 25 Effect of stability factor, f=0.5, at switch 2

The simulation results show that with long control delay, a large stability factor f does not necessarily converge to the target faster. To have best performance, in practice the control gain α needs to be fine-tuned based on formula (5).

4.2 Supporting rt-video Traffic over ABR

The following simulation experiments show how the FASTRAC calculates fair shares based on the measurements of background traffic load and ABR traffic load at the bottleneck switch. The available bandwidth for ABR is allocated among the ABR sources based on the fairness allocation scheme and weight assigning method. We run the simulations under different network configurations to evaluate the performance of ABR services when supporting rt-video traffic with the FASTRAC rate control algorithm. First we consider the Bottleneck configuration with only two ABR sources supporting video traffic. Based on the simulation results, we introduce a dynamic weight fairness allocation scheme in order to eliminate the problems of the fixed weight scheme when supporting rt-video application. We also show the improvement of system performance when ABR sources are added to support greedy traffic. Then we extend our study to the Generic network configuration with different control delays.

4.2.1 Bottleneck Configuration

As shown in Figure 5, for simplicity, the Bottleneck configuration consists of two ABR ATM switches, one VBR video source as the aggregate background traffic, and two MPEG-2 video sources as the ABR end systems. The configuration is as follows:

Switch 1 and switch 2:

ATM explicit rate algorithm: the FASTRAC rate control

Bottlenecked Link Capacity: 20 Mbps

Target Link Utilization: 95%

Link Delay: 0.01 second

Control delay = 3.2E-5 second.

Traffic Descriptions and Parameters:

(1) VBR background traffic

We use Self-Similar traffic to simulate the statistical multiplexing of real-time variable-bit-rate video traffic, as shown in Figure 26 as an example. The refresh interval, maximum rate and mean rate are specified below.

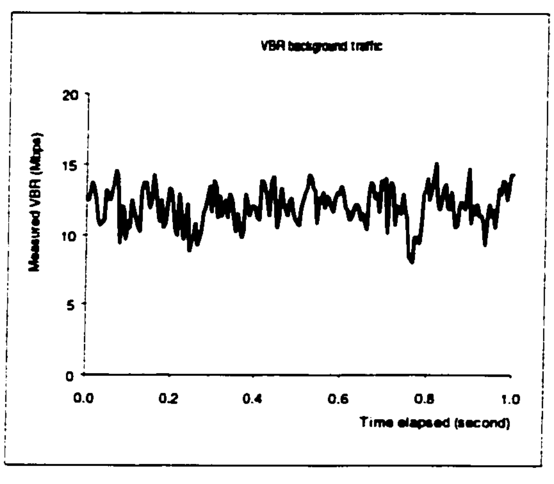


Figure 26 Self-Similar VBR background traffic

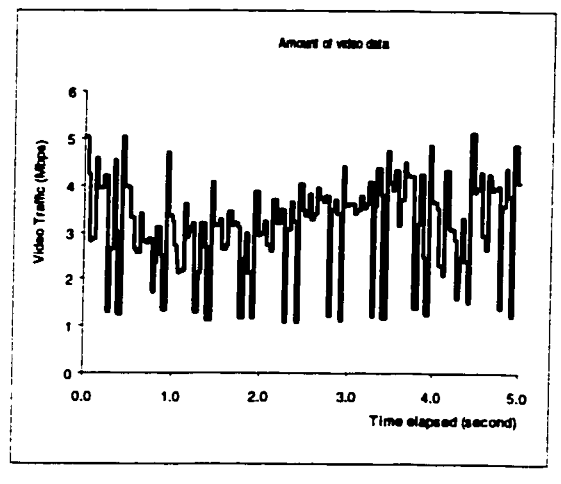


Figure 27 MPEG-2 rt-video traffic

(2) ABR source

MPEG-2 compressed video with rate statistics as shown in Figure 27 (as an example):

The peak video rate = 5.11 Mbps

The minimum video rate = 1.07 Mbps

PCR = 20 Mbps

Access link delay =1.6E-5

(1) ABR service transporting greedy traffic

Before we examine the scenarios with rt-video traffic, let us first have a look at the case of greedy traffic. The set of figures below show the results of the Bottleneck configuration with all ABR sources transmitting greedy traffic. Here we use a *Max-Min* fairness scheme with the two ABR sources having the same weight. The result in Figure 28 shows that the computed explicit rate reflects the changing condition of the VBR background traffic, i.e., when the background VBR traffic increases, the ABR fair share decreases accordingly, and vice versa. Figure 29 and Figure 30 depict the available ABR

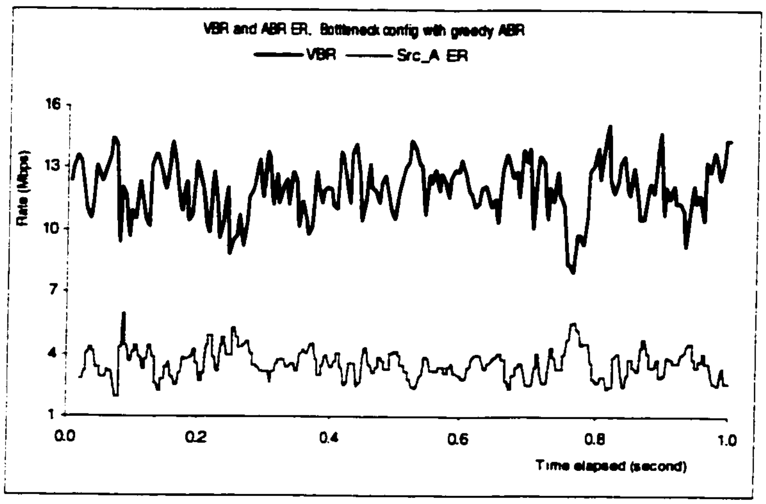


Figure 28 ABR fair share vs. background traffic. Bottleneck with greedy traffic

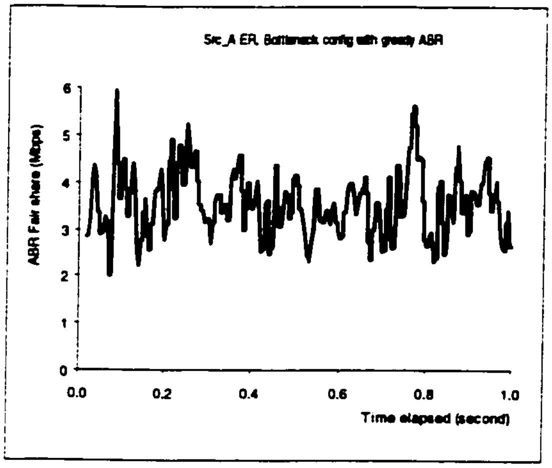


Figure 29 Src_A fair share, Bottleneck with greedy traffic

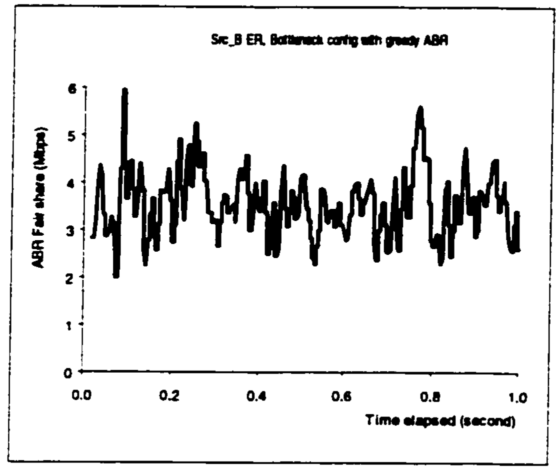


Figure 30 Src_B fair share, Bottleneck with greedy traffic

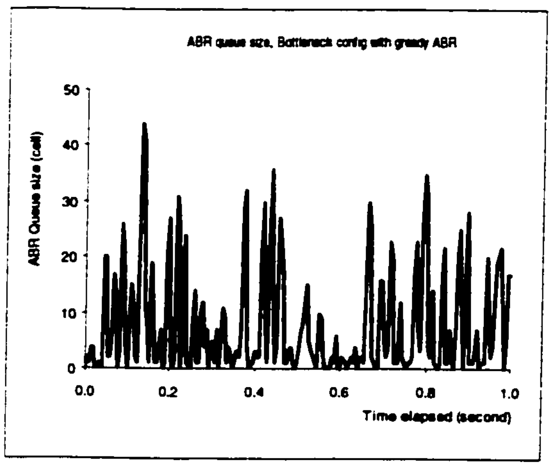


Figure 31 ABR queue size, Bottleneck with greedy traffic

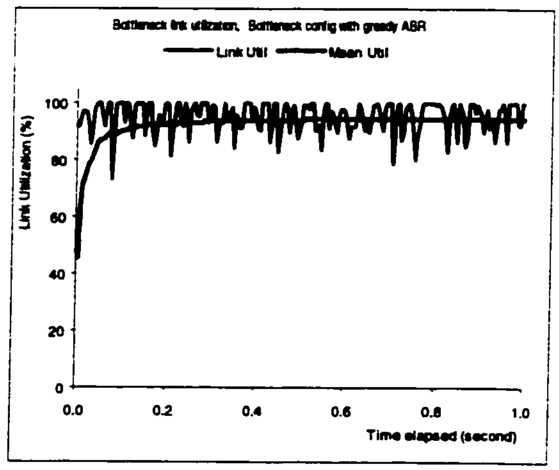


Figure 32 Link utilization, Bottleneck with greedy traffic

bandwidth is equally allocated between two ABR sources. Each ABR greedy source can adjust its transmitting rate exactly to the allowed cell rate. The link utilization of the bottleneck switch (as shown in Figure 32) is very close to the target, since the bottleneck switch is fully loaded at most of the time. When there is a rate increase on the background traffic, the ABR queue will accumulate a small number of ABR cells. As shown in Figure 31, the ABR queue size is bounded to a very small size because the FASTRAC switch is dynamically adjusting the fair shares according to the current VBR background traffic measurement as well as the ABR queue size. With ABR sources supporting greedy traffic, it is an extreme case that ABR sources perfectly adapt their rates to the explicit rates feedback from the network.

(2) ABR supporting Video traffic with fixed weight (Max-Min fairness criteria)

When ABR sources are transmitting greedy traffic, ABR sources can ideally adapt to the changes of the available bandwidth of the network. The application of file transfer is an example of greedy traffic. When ABR sources are used to transport rt-video traffic, due to the nature of video traffic and its requirement, rt-video ABR sources are not so adaptive as greedy ABR sources. The following simulation scenario is to evaluate the performance of the FASTRAC switch when supporting rt-video traffic. We still consider a *Max-Min* fairness rate allocation scheme. The ABR sources are transmitting rt-video clips as shown in Figure 27 above. The ABR source adapts to the explicit rate by chopping the video data that exceeds the allowed cell rate. The simulation results below illustrate the behavior of the system when supporting rt-video traffic with fix equal weight.

As shown in Figure 33, the ABR explicit rates change accordingly when the background traffic on the bottleneck link changes. Figure 34 and Figure 35 show that the fair shares for the two ABR sources are identical although the video traffic distribution patterns of source A (Src_A) and source B (Src_B) are different. This complies with the *Max-Min* fairness allocation criteria and fixed equal weight assignment for source A and source B.

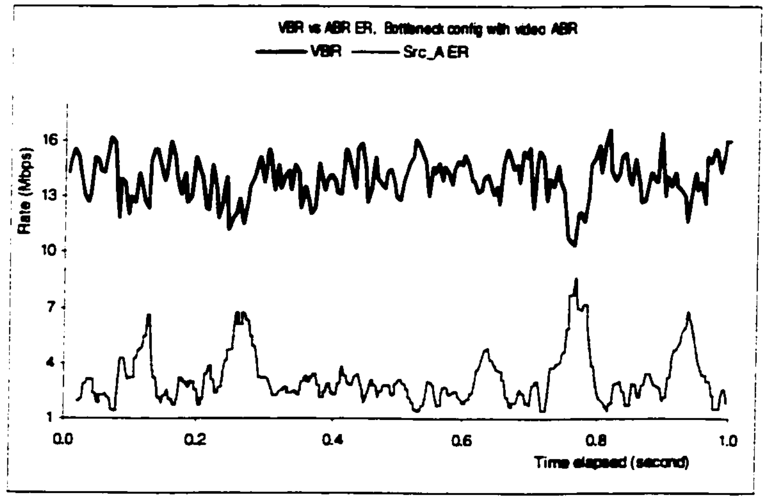


Figure 33 ER fair share vs. background VBR, Bottleneck with video traffic

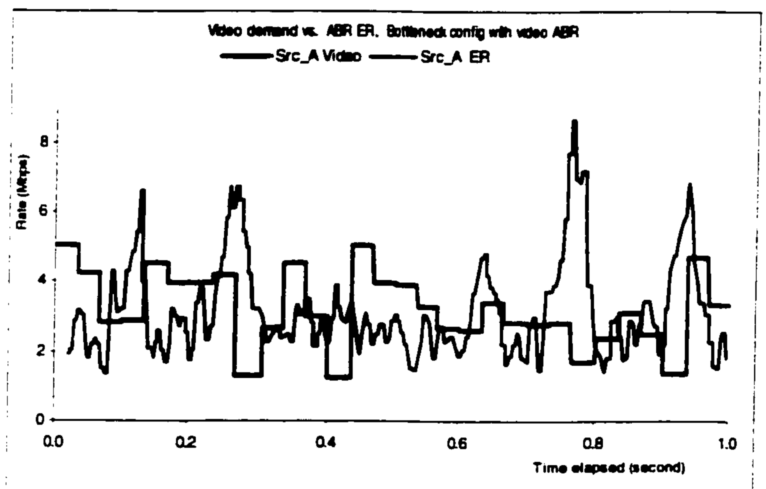


Figure 34 Src_A ER vs. demand, Bottleneck with video traffic

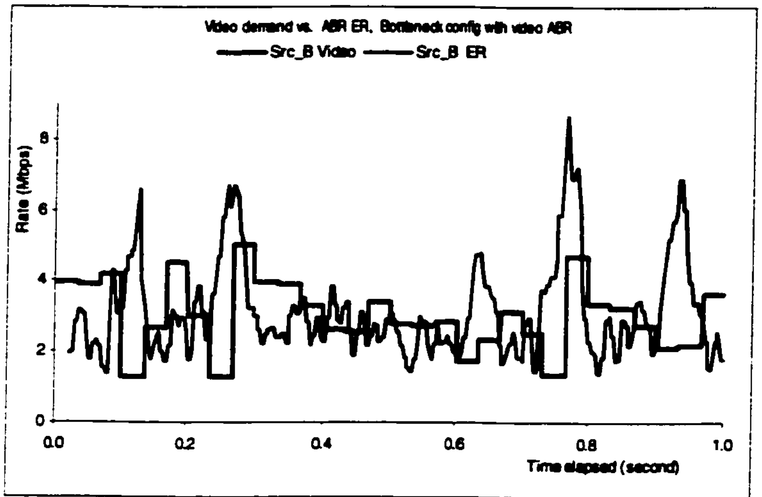


Figure 35 Src_B ER vs. demand, Bottleneck with video traffic

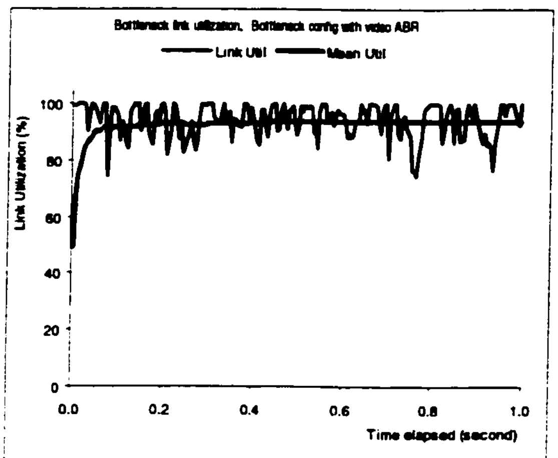


Figure 36 Link utilization, Bottleneck with video traffic

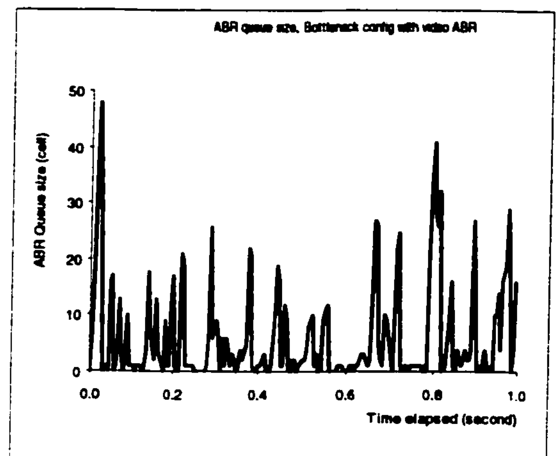


Figure 37 ABR queue size, Bottleneck with video traffic

Figure in 36 illustrates the link utilization and the average utilization of the bottlenecked link. The average link utilization for the one-second simulation is 93.8% which is lower than the average link utilization when ABR sources are transmitting greedy traffic (94.8%, Figure 32). Figure 37, 38 and 39 depict respectively the ABR queue size at the bottleneck switch (switch 1) and the Cell Transfer Delay (CTD) of ABR source A and B. Both ABR queue size and cell transfer delay (CTD) are bounded within acceptable values. The cell delay variation is introduced by the differentiation of the queuing delay of ABR cells at the bottleneck switch. The ABR queue behavior is the same as in the case of greedy traffic – when the VBR background traffic suddenly increases its rate, there will be an ABR queue size at the bottleneck switch. The accumulated ABR cells in the queue are drought up with the FASTRAC reducing the fair shares for the ABR sources.

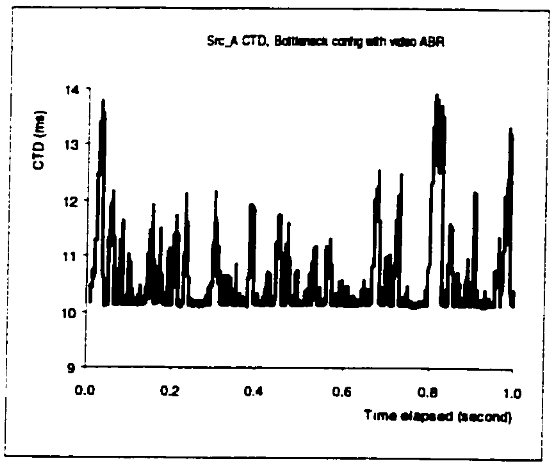


Figure 38 Src_A CTD, Bottleneck with video traffic

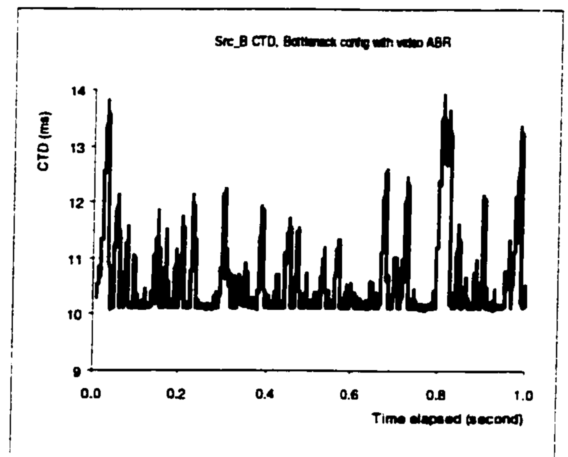


Figure 39 Src_B CTD, Bottleneck with video traffic

As we have described above the ABR source adaptation behavior, the actual cell rate of the ABR source is constrained by the following formula set:

$ACR = min\ \{ACR + RIF * PCR, PCR\}$,

$ACR = min\ \{ACR, ER\}$,

$ACR = max\ \{ACR, MCR\}$,

$Actual\_ABR\_Cell\_Rate = min \{ACR, Video\_Rate\_Demand\}$

As depicted in Figure 34 and Figure 35, sometimes the video source does not generate enough traffic to fully occupy the available bandwidth. This means that there is no enough traffic for the source to fully-load the bottleneck link. That's why the average link utilization with video traffic (Figure 36) is lower than that with greedy traffic (Figure 32).

We can describe this behavior by introducing the concept of ABR switch load state. ABR switch has three states according to its traffic load, namely, "under-load" state, "fully-load" state and "overload" state. The "overload" state is in fact only a transit state. Once the switch becomes overloaded, it will be transiting to "full load" state by decreasing ABR fair shares. To analyze the conditions of these states and the transition between them, let us consider the following three scenarios:

(1) The fair share of ABR source is larger than its video demand:

The switch will be in the "under-load" state and each ABR source can transmit all its video traffic. Because the aggregated ABR traffic transmitted can not reach the available bandwidth for ABR sources (ABR target $T(n)$), the measured ABR traffic load ($M(n)$) is less than the target. As a result, the FASTRAC will keep increasing the fair shares until either the fair share reaches its upper bound, i.e., PCR, or transit to the "fully-load" state whichever is applicable.

(2) The ABR source's video rate is larger than or equal to its fair share:

Each ABR source can fully utilize its fair share credit and the bottlenecked link is fully utilized up to the target. The switch will remain at the "fully-load" state until either the VBR traffic changes or the ABR video rate changes. The ABR video information that exceeds the fair share will be dropped. The information dropped is not proportional to the corresponding video frame size of the ABR sources. For example, if one ABR source is transmitting an I-frame while another is transmitting a B-frame, more percentage of information of the I-frame will be dropped which in fact is more important to the QoS of the video. The policy of dropping data is not reasonable.

(3) The fair share is larger than the video rate of one ABR source but smaller than the video rate of another:

In this scenario, the switch is initially at "under-load" state. Since the total ABR traffic of these two ABR sources is less than the total available fair shares, the fair share will be increased so the ABR source with more video data can transmit more. The fair share will keep increasing until either the fair share reach its upper bound (i.e., PCR) or transit to "fully-load" state. In such a way, the unused fair share credit of one source can be used by aother source.

From the simulation and analysis, we can conclude:

The computed ABR fair shares reflect the changing of the VBR background traffic. The unused bandwidth of one ABR source will be used by other ABR sources. This is one of the advantages of the FASTRAC rate control algorithm. It is particularly important when supporting real-time video transmission over ABR connection. Due to the control delay, RM cell interval and load measurement interval, it takes time for the bottlenecked ABR switch to increase the fair shares. As a result, during the fair share adjustment time, one ABR source may have to drop video data due to inadequate fair share, while in fact there is still available bandwidth not being used by other sources. This is one of the problems of the FASTRAC rate control when transmitting real-time video traffic using fixed weight allocation scheme.

Another problem with fixed weight allocation is that the dropping of the exceeded video data is not fair among the ABR sources. Because the fair share allocation algorithm does not take into account the ABR source's bandwidth demand, the data dropping policy is not for the benefit of all the ABR sources. To overcome the problems, we introduce a dynamic weight fairness allocation method.

(3) ABR supporting Video traffic with dynamic weight scheme

(1) Dynamic weight fairness allocation:

In order to solve the problems incorporated with the fixed weight method, we introduce a dynamic weight method. The objective of the proposed scheme is to allocate the available bandwidth resource among the ABR sources according to their current demands. In the case of supporting rt-video traffic, the current video rate is used as the weight of each ABR source when allocating the total available bandwidth. The fair share computation and allocation are the same as in the fixed weight experiments above, except the weight is the current video rate and is changing dynamically. Two actions are taken to improve the efficiency and responsiveness:

- The current bandwidth demand is sent to the switches via the CCR field of a RM cell of each ABR source, and the bandwidth request is constrained by its peak cell rate (PCR) that is specified in the Admission Contract so that the CCR cheating problem can be avoided;
- In order to reduce the effect of propagation delay of RM cells, the CCR information is derived from the forward RM (FRM) cells instead of the BRM cells. So the weight being used can reflect the most recent fair share demand of the ABR source.

(2) Simulation experiments

With the same configuration and parameters used in the simulation with the fixed weight scheme, Figure 40 and Figure 41 show the results of ABR fair share allocation with the proposed dynamic weight allocation method being used. The results show that the total available bandwidth is allocated among the ABR sources proportionally to their video rates which are used as the weights for the ABR connections. The more data a source has to send, the more bandwidth it will get. This is particularly useful when transmitting MPEG-2 video traffic. For example, when one ABR source has an I-frame to transmit while another ABR source has a B-frame to transmit, the ABR source with an I-frame will get more bandwidth and hence can transmit more. As an

I-frame usually contains more important information. In this way, the QoS of all the ABR applications is balanced in a fair way, avoiding unfairness and severe QoS variation among the ABR connections.

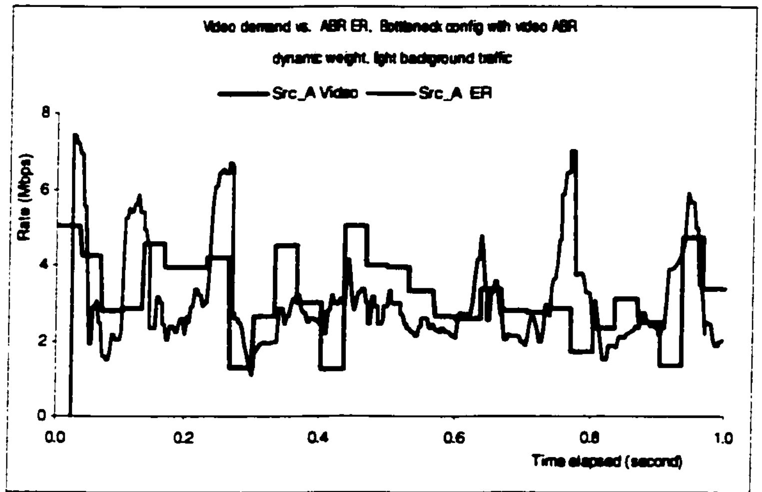


Figure 40 Src_A ER vs. demand, Bottleneck, light, dynamic weight

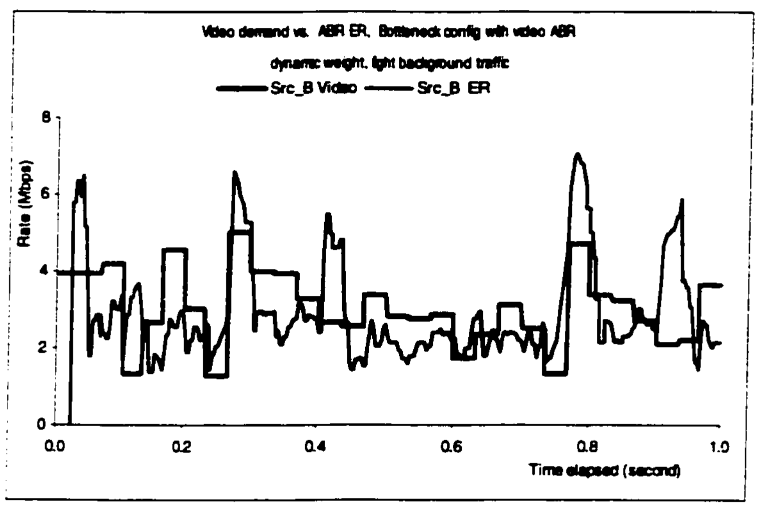


Figure 41 Src_B ER vs. demand, Bottleneck, light, dynamic weight

The result above is the case when background traffic is light. It also works well when the background traffic is heavy. Figures 42 to 45 show how the dynamic weight scheme improves the fair share allocation among the ABR sources compared to the fixed weight scheme.

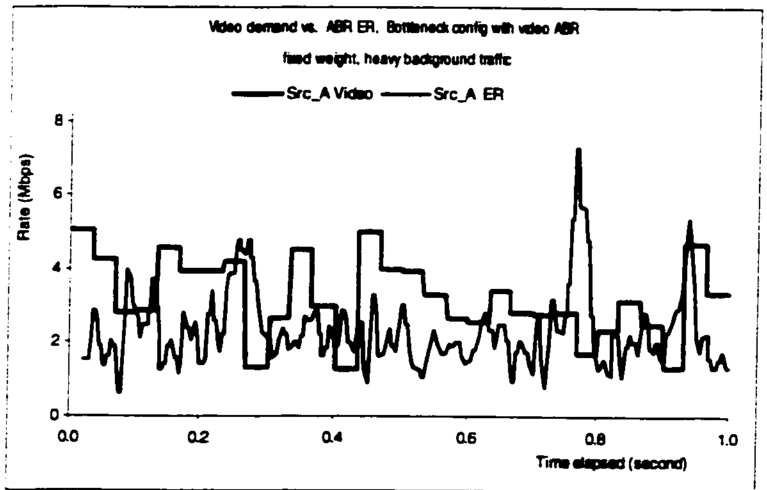


Figure 42 Src_A ER vs. demand, Bottleneck, heavy, fixed weight

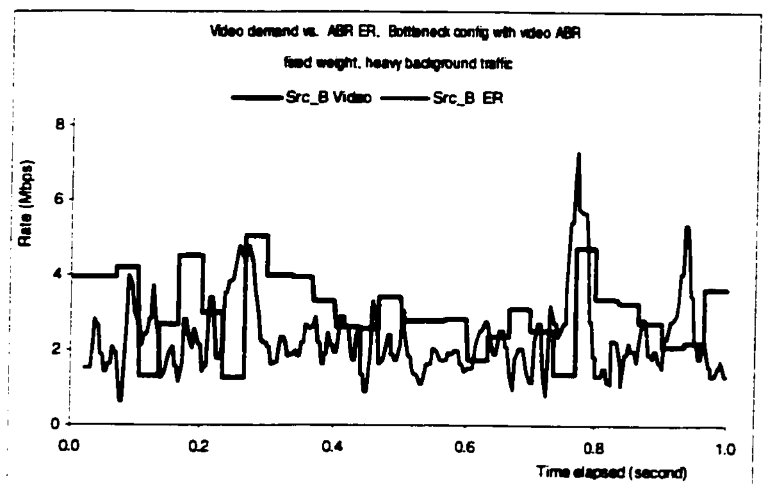


Figure 43 Src_B ER vs. demand, Bottleneck, heavy, fixed weight

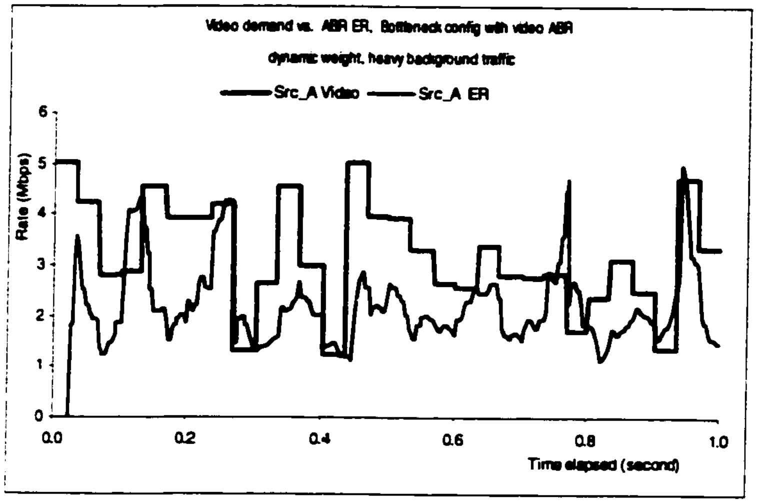


Figure 44 Src_A ER vs. demand, Bottleneck, heavy, dynamic weight

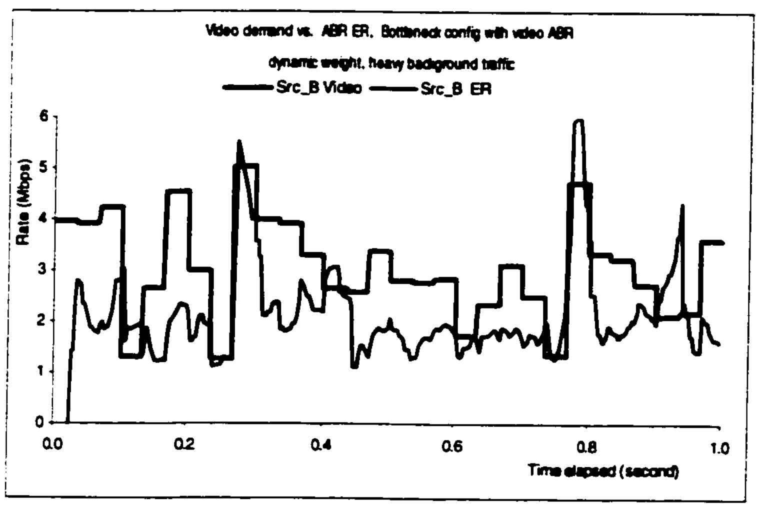


Figure 45 Src_B ER vs. demand, Bottleneck, heavy, dynamic weight

The simulation results above show that in both scenarios, the dynamic weight method can improve the fair share allocation. The dynamic weight fair share method can dynamically determine the weights of the ABR sources according to their demands and the total available fair shares are allocated to different ABR sources according to their weights. The performance is improved compared to the fixed weight method, which can not reflect the time-varying source demand.

Although the performance is improved, some of the bandwidth resources are being wasted when there is no enough video traffic on the link, especially in the scenario of light background traffic as shown in Figures 40 and 41. To utilize the unused available bandwidth that will be wasted otherwise, we add more ABR sources to transmit non real-time applications, such as file transfer, which is not sensitive to delay variation.

(4) Supporting both real-time video and non real-time applications

(1) Network Configuration

In order to utilize the unused bandwidth that will be wasted otherwise, we add two more ABR sources as shown in Figure 46. Src_A and Src_B still support rt-video traffic and have higher priority than the added ABR sources. The two added ABR sources (Src_C and Src_D in Figure 46) support non real-time application characterized as greedy traffic, such as file transfer. They will occupy the additional bandwidth that is not fully utilized by the video sources (Src_A and Src_B). We still use the video rates of SrC_A and Src_B as their dynamic weights. The spare bandwidth is allocated equally among the added greedy sources according to the following formula for weight calculation:

$$W_j(n) = [T(n) - D_v(n)]/N \qquad \text{for all } j \qquad (6)$$

Where:

$W_j(n)$ is the weight of ABR greedy source j

T(n) still represents the target total available fair share

$D_v(n)$ is the total bandwidth demand of ABR video sources

N is the total number of greedy ABR sources

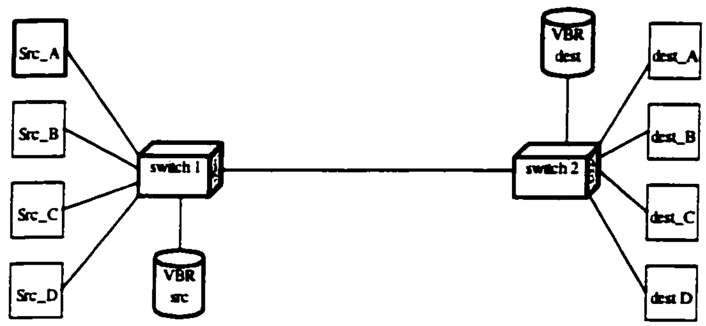


Figure 46 Bottleneck configuration with both video and greedy ABR sources

(2) Simulation results

Figure 47 to Figure 50 show the explicit fair shares for the four ABR sources. Figure 47 and 48 illustrate the Explicit Rates (ER) and video data rates of Src_A and Src_B that

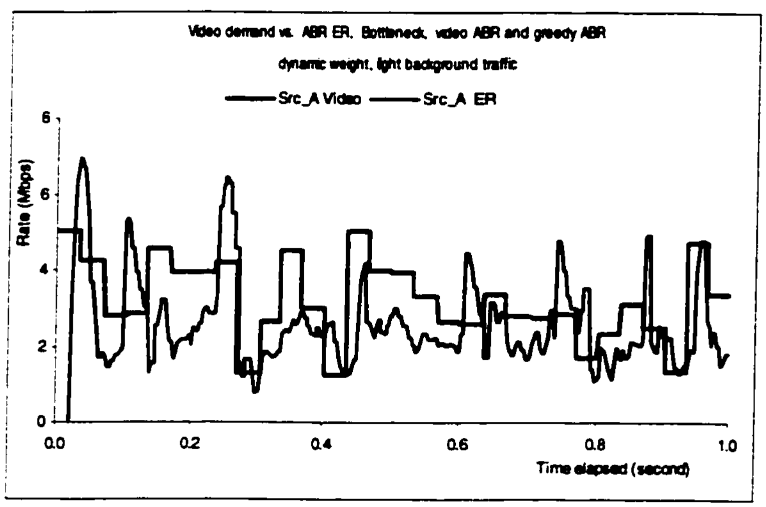


Figure 47 Src_A ER vs. demand, Bottleneck, video and greedy

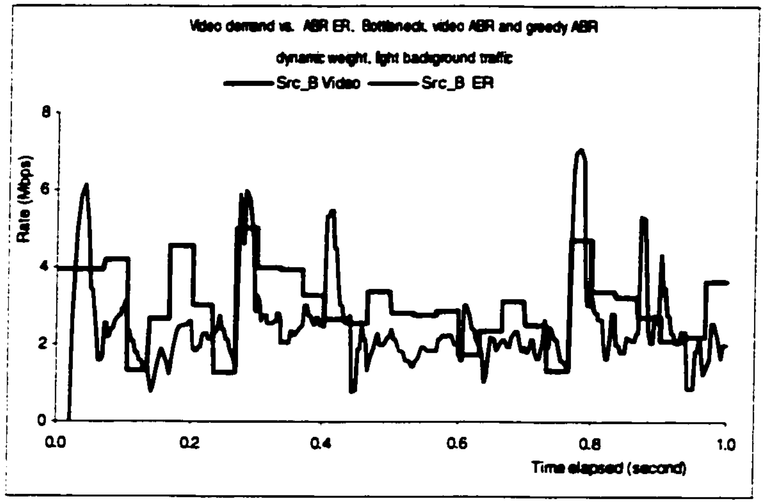


Figure 48 Src_B ER vs. demand, Bottleneck, video and greedy

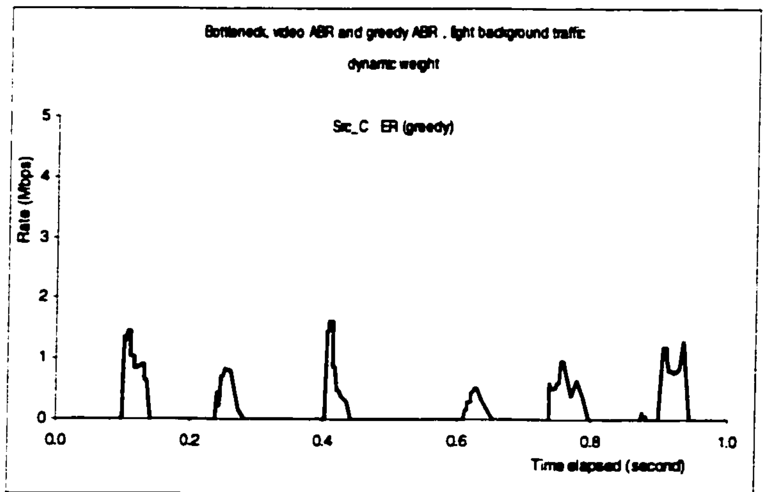


Figure 49 ER for greedy Src_C, Bottleneck, video and greedy

are supporting video traffic. The fair share allocated to the video source is proportional to the video rate of each ABR source. The fair shares for the two video sources are not affected by the two greedy sources. The additional available bandwidth left unused by Src_A and Src_B is occupied by Src_C and Src_D as shown in Figure 49 and Figure 50.

Another observation is that the additional bandwidth is allocated equally between Src_C and Src_D. It reflects the equal weight scheme in (6).

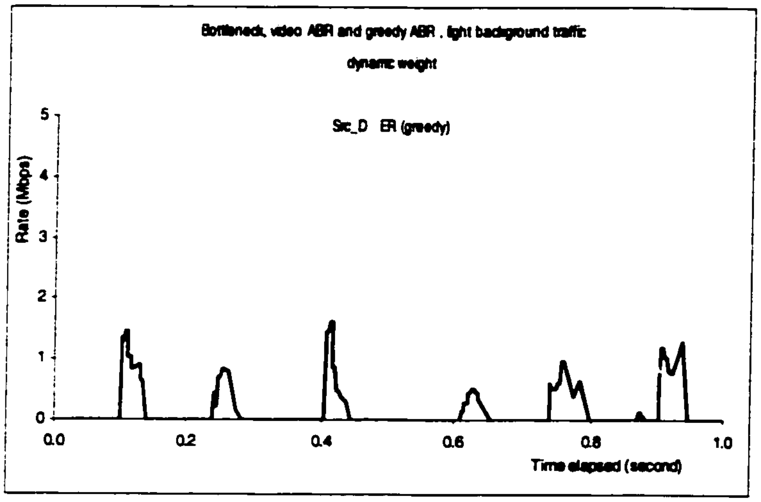


Figure 50 ER for greedy Src_D, Bottleneck, video and greedy

4.2.2 Generic Fairness Configuration

In the Bottleneck configuration, all the ABR sources are very close to the bottleneck switch. The control delay is very small. In the above simulations with the Bottleneck configuration, the access link delay from ABR source to the bottleneck switch (switch 1 in Figure 5 and Figure 46) is 1.6E-5 second. Thus the control delay is only 3.2E-5 second. The explicit rate each ABR source received reflects the current network load condition, and also the rate adaptation of the ABR sources takes effect at the bottleneck switch only after a very short delay (1.6E-5 second). In real networks, usually the bottleneck switch is not close to all ABR sources. The control delays of the ABR sources are different. The following simulations will examine the impact of control delay on the responsiveness and performance of ABR service supporting rt-video using the FASTRAC rate control algorithm.

We consider Generic configuration in the following simulation experiments. As shown in the network diagram in Figure 6, VBR_src generates Self-Similar traffic as the VBR background traffic at switch 3. Bottleneck link is at switch 3. Src_A, Src_B and Src_C support rt-video traffic and other ABR sources (Src_D, Src_E and Src_F) support non real-time applications, such as file transfer and email, which are characterized as greedy traffic. There are three ABR connections on the bottleneck link (between switch 3 and switch 4) with Src_A, Src_B and Src_C as sources. The control delays for these three ABR connections are different. On other links, there are both video source connection(s) and greedy source connection. Here we only consider the performance of the ABR sources supporting rt-video traffic.

(1) Fixed weight allocation with long control delay

The set of figures below show the simulation results of the Generic configuration with link delay of 10 ms and fixed weight allocation. Figure 51 illustrates how ABR explicit rate for Src_A follows the changing of the background traffic using fixed weight scheme. From the simulation result we can see the computed fair share still follows the rate changes of the background VBR traffic, but with a delay of approximately 30 ms. The delay is the propagation delay the BRM cells experienced when transmitting from the bottleneck switch to Src_A.

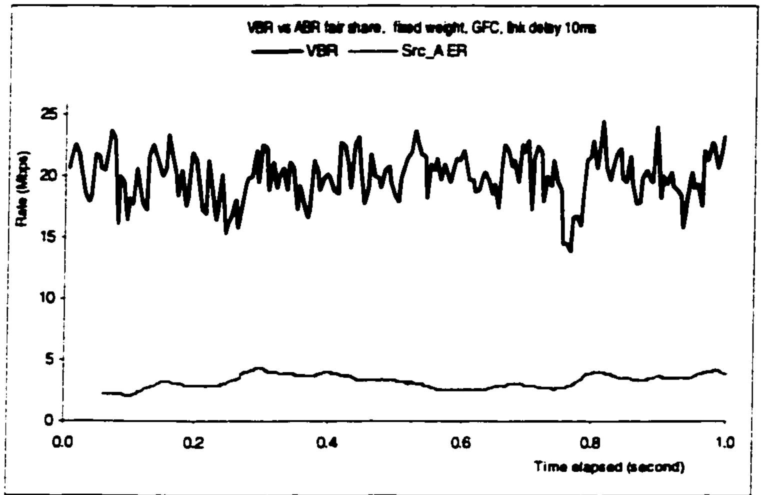


Figure 51 ER vs. VBR traffic, GFC, fixed weight, d=10ms

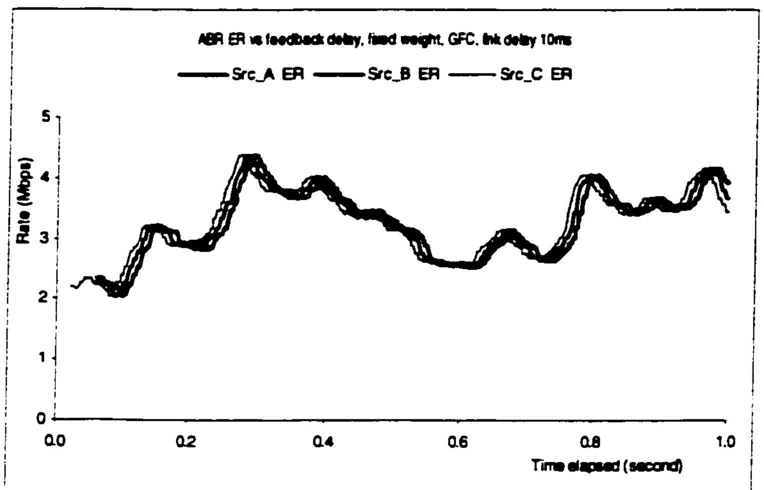


Figure 52 ER vs delay, GFC, fixed weight, d=10ms

The fair shares for Src_B and Src_C are the same as for Src_A. They all follow the changing of the VBR background traffic at switch 3 but with different feedback delay due to difference of the distance to the bottleneck switch. As shown in Figure 52, it takes longer (about 20 ms) for Src_A to get its explicit rate than Src_C and Src_B, since Src_A is two hops farther to the bottleneck switch (switch 3) than Src_C. Src_B gets its feedback from the network about 10 ms earlier than Src_C. The backward RM cells carrying the explicit rate feedback information for Src_A take about 20 ms to arrive Src_A, and Src_A adapts its rate according to the network feedback. The rate adaptation takes effect at the bottleneck switch after another 20 ms. By the time the feedback for Src_A starts taking effect at the bottleneck switch, the network condition already changed. Thus the response is no longer accurate and this will increase the ABR queue size, as shown in Figure 53.

Figures 54, 55 and 56 show the ER fair shares of Src_A , Src_B and Src_C respectively. As we use fixed weight scheme for the fair share allocation, the ER fair share allocated to each ABR source does not reflect the transmission demand of its video traffic.

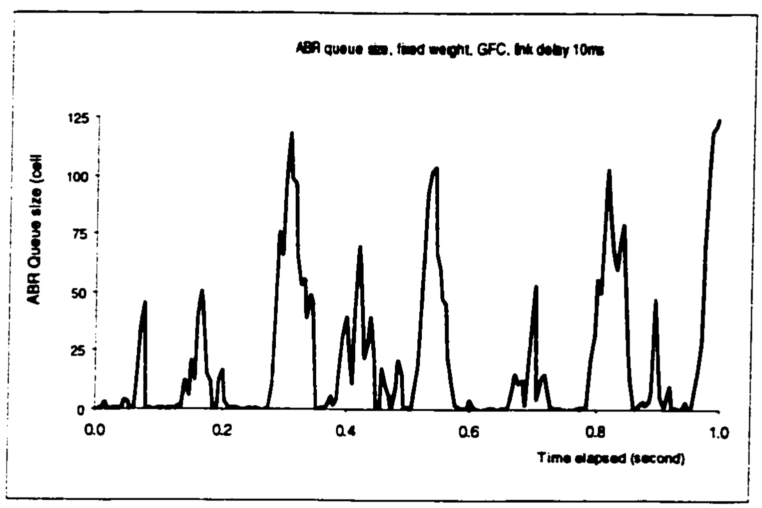


Figure 53 ABR queue size, GFC, fixed weight, d=10ms

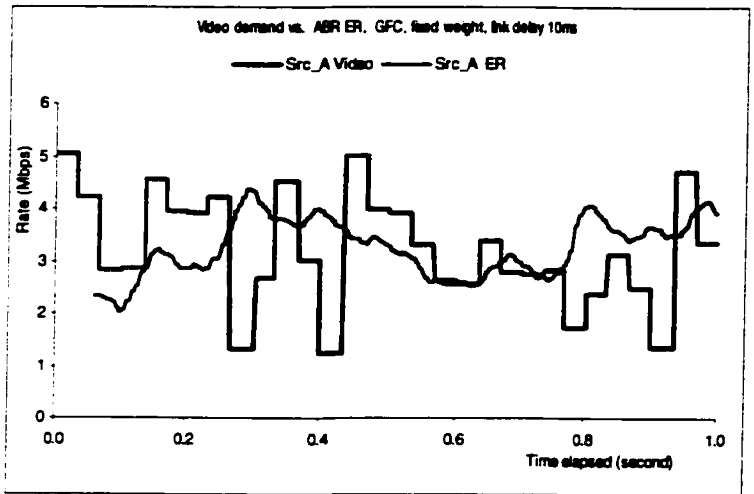


Figure 54 Src_A ER vs. demand, GFC, fixed weight, d=10ms

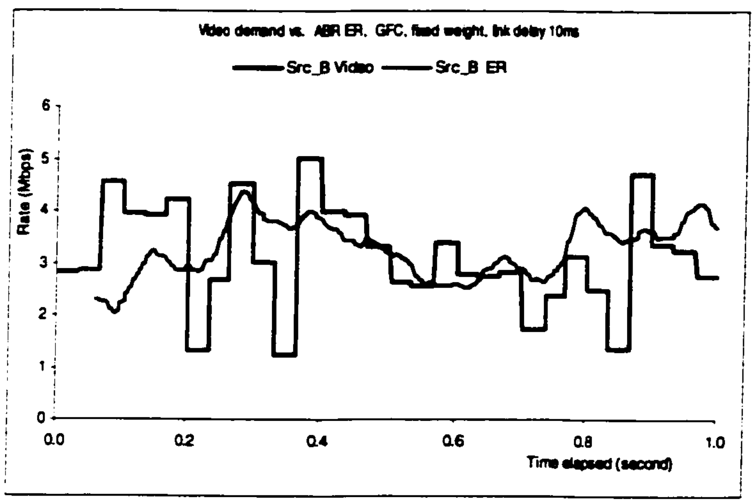


Figure 55 Src_B ER vs. demand, GFC, fixed weight, d=10ms

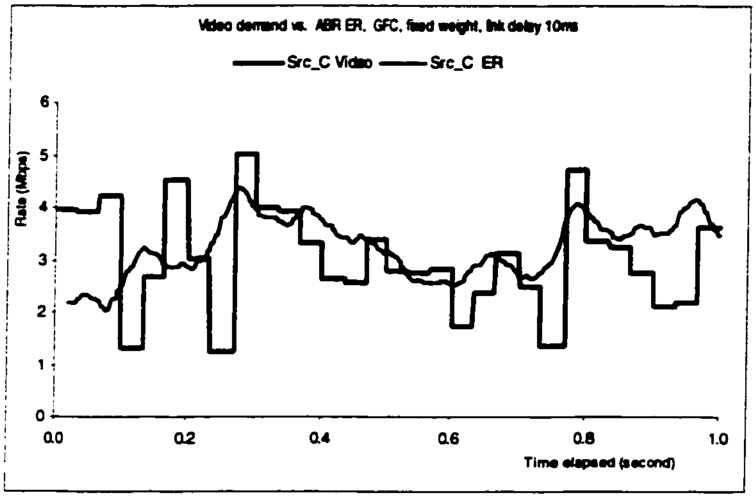


Figure 56 Src_C ER vs. demand, GFC, fixed weight, d=10ms

(2) Dynamic weight allocation with long control delay

Now we consider the dynamic weight scheme with the same network configuration and parameters as in the simulation experiment above with the fixed weight allocation. Figure 57 to Figure 62 As show the simulation results when using dynamic weight scheme. As shown in Figure 57, 60, 61 and 62, the fair share for each ABR source depends on both the background VBR traffic load and the bandwidth demand of each source. Figure 58 illustrates the impact of feedback delay on the explicit rate of the ABR sources. The feedback delay difference among the sources is still the same as in the experiment with fixed weight scheme, but the fair shares for Src_A, Src_B and Src_C are no longer the same, which is an expected result with the dynamic weight scheme being considered. Besides the impact of fair share feedback delay and control delay that are common to both fixed weight scheme and dynamic weight scheme, there is an additional problem with the dynamic weight scheme. Because of the long forward propagation

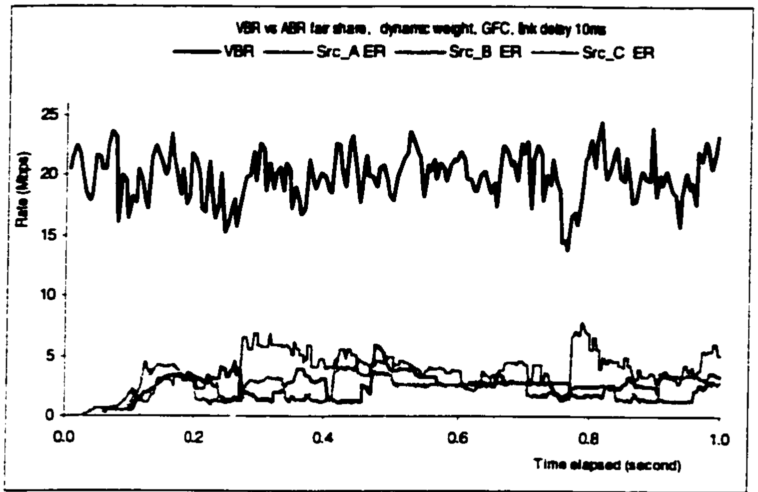


Figure 57 ER vs. VBR traffic, GFC, dynamic weight, d=10ms

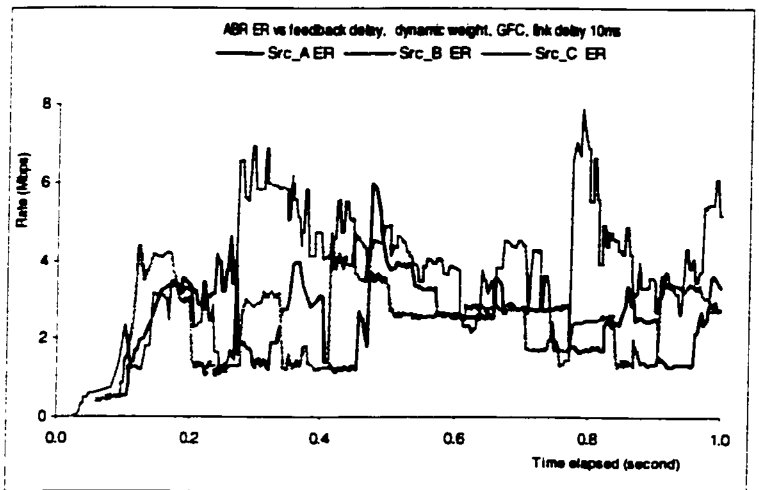


Figure 58 ER vs. delay, GFC, dynamic weight, d=10ms

delay from Src_A and Src_B to the bottleneck switch (switch 3), the dynamic weights received at the bottleneck switch do not reflect the current fair share demands of the ABR Src_A and Src_B. They were the fair share demands long time ago – 20 ms ago for Src_A and 10 ms ago for Src_B. Along with the long feedback delay and control delay aforementioned, the dynamic weight scheme makes the responsiveness performance even worse due to the additional error introduced by the delayed weight information of the ABR sources.

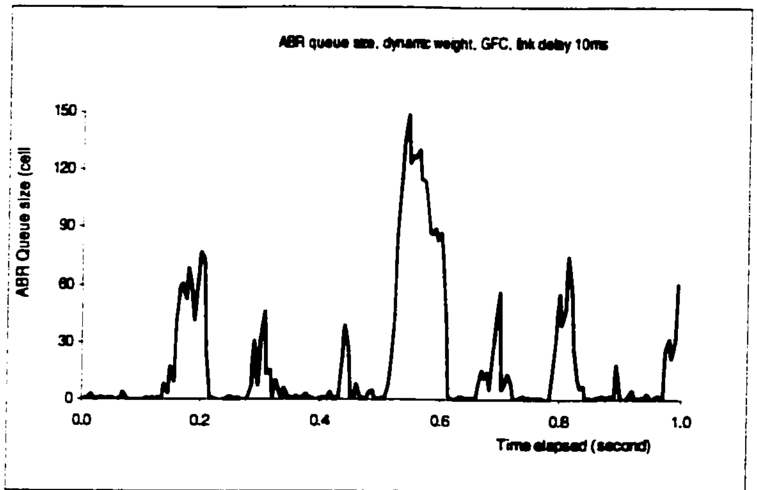


Figure 59 ABR queue size, GFC, dynamic weight, d=10ms

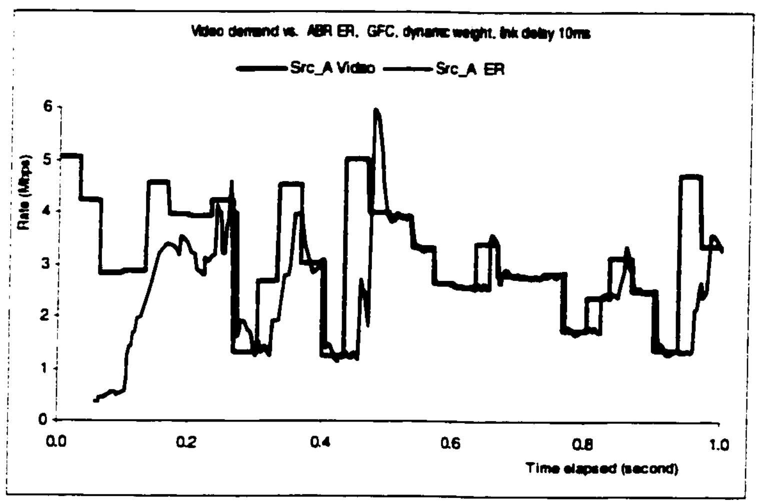


Figure 60 Src_A ER vs. demand, GFC, dynamic weight, d=10ms

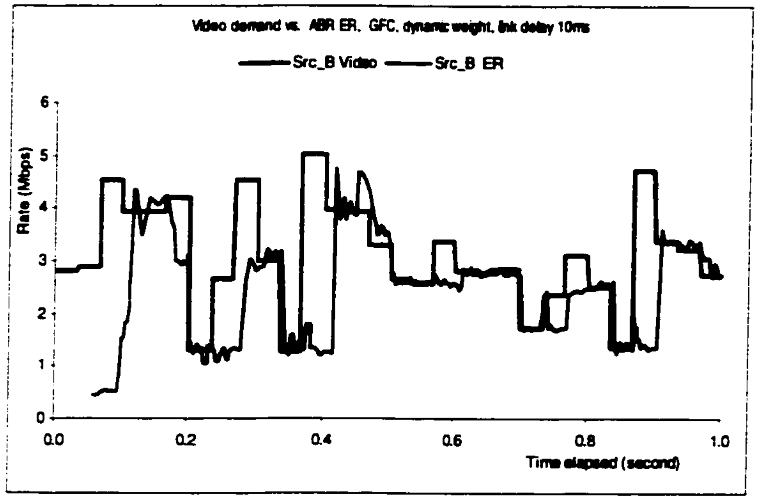


Figure 61 Src_B ER vs. demand, GFC, dynamic weight, d=10ms

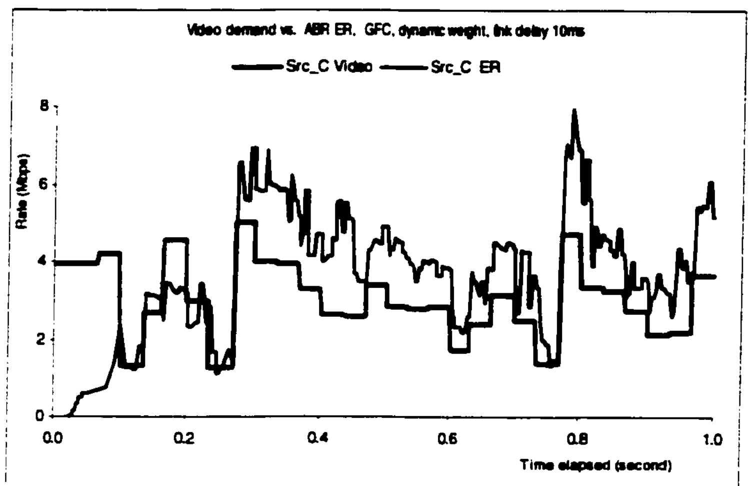


Figure 62 Src_C ER vs. demand, GFC, dynamic weight, d=10ms

(3) Fixed weight allocation with short control delay

Let us look at the case of small control delay with the same Generic configuration. Now let us reduce the link delay between switches to 1 ms. Figures 63 to Figure 68 below illustrate the results of the simulation experiments. As expected, the fair share for the

ABR source follows the available bandwidth of the bottleneck link very well. Since there is only a very short delay among the feedback for each ABR source, the condition is very close to the aforementioned Bottleneck configuration in the sense of the control delay and feedback delay.

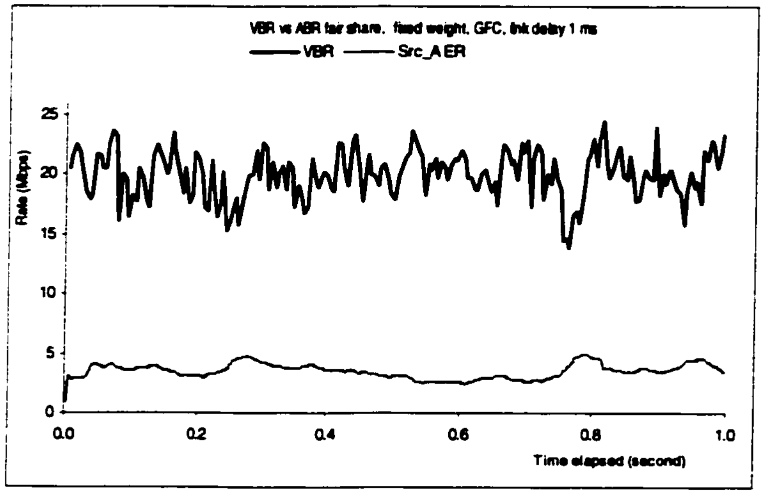


Figure 63 ER vs. VBR traffic, GFC, fixed weight, d=1ms

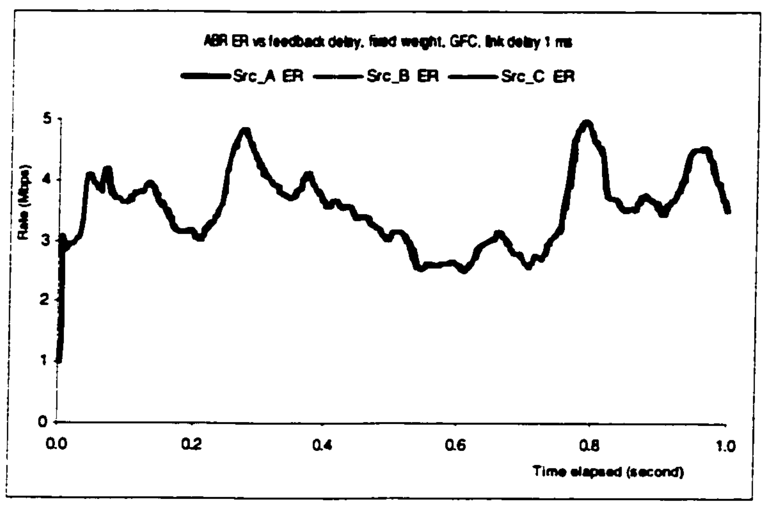


Figure 64 ER vs delay, GFC, fixed weight, d=1ms

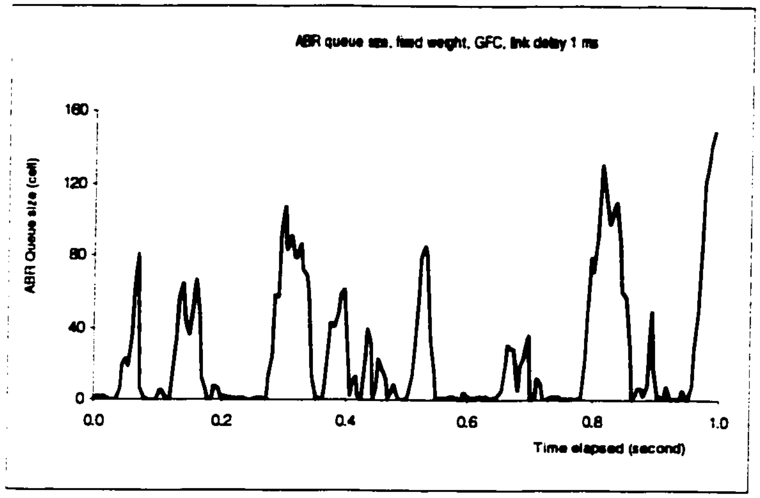


Figure 65 ABR queue size, GFC, fixed weight, d=1ms

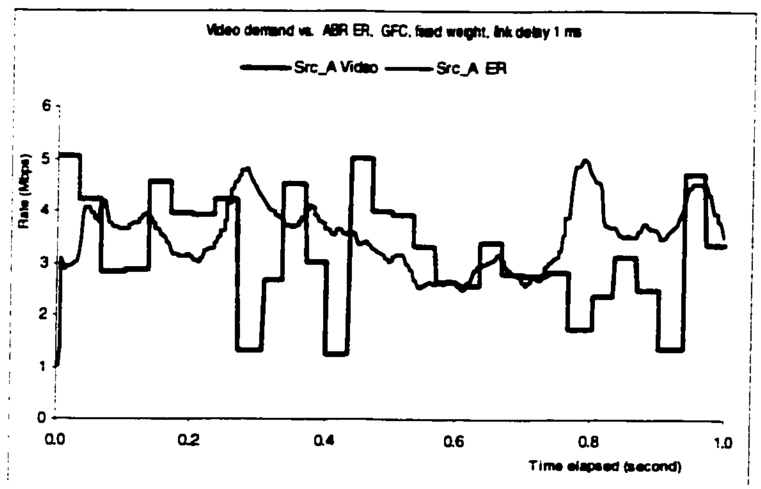


Figure 66 Src_A ER vs. demand, GFC, fixed weight, d=1ms

As shown in Figure 66, 67 and 68, the fair share for each ABR source does not reflect the its bandwidth demand. The fair share allocation among the ABR sources is not for the benefit to improve the overall QoS of the transmitted rt-video traffic. Some ABR source does not have enough to send while others don not have enough bandwidth. The dropping of the video data is not reasonable either. In order to improve the fair share allocation among the ABR sources, we use dynamic weight allocation scheme in the following simulation experiments.

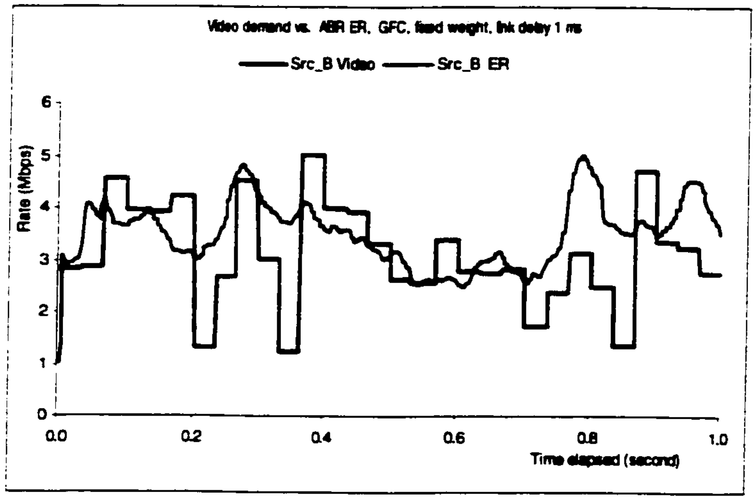


Figure 67 Src_B ER vs. demand, GFC, fixed weight, d=1ms

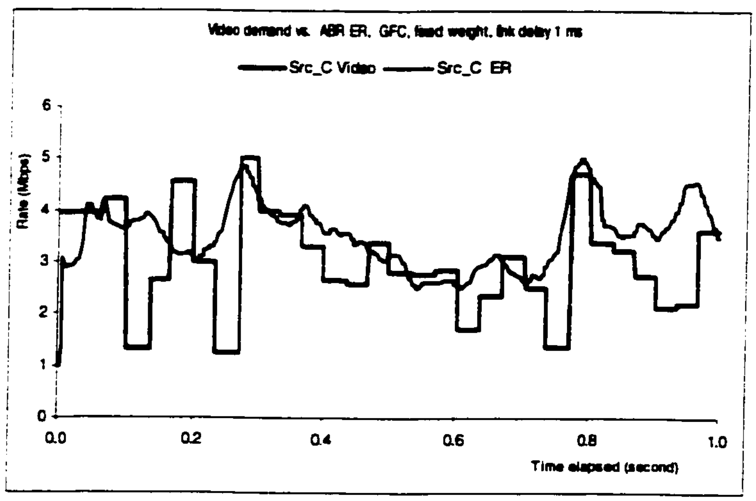


Figure 68 Src_C ER vs. demand, GFC, fixed weight, d=1ms

(4) Dynamic weight allocation with short control delay

With the same configuration and parameters as in the above simulation, here we use the dynamic weight allocation scheme. The simulation results demonstrated that the fair shares for the ABR sources follow the changes of the background traffic and they also reflect the bandwidth demand of each ABR source. Figure 69 to Figure 74 illustrate the improvement of the fair share allocation compared to the fixed weight scheme.

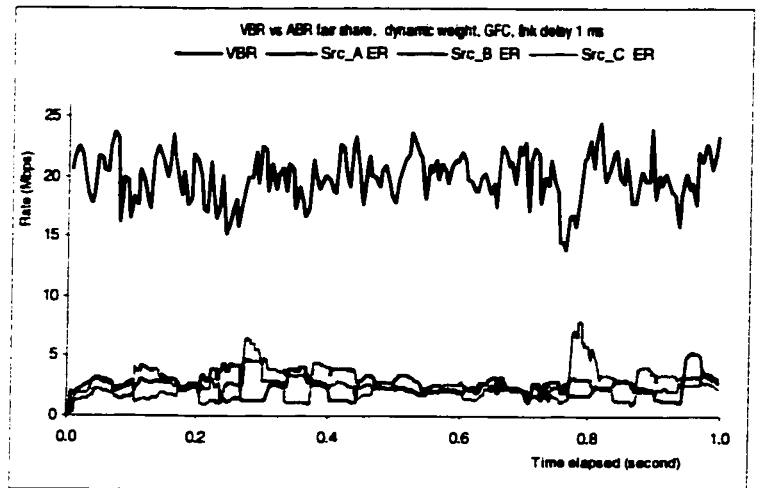


Figure 69 ER vs. VBR traffic, GFC, dynamic weight, d=1ms

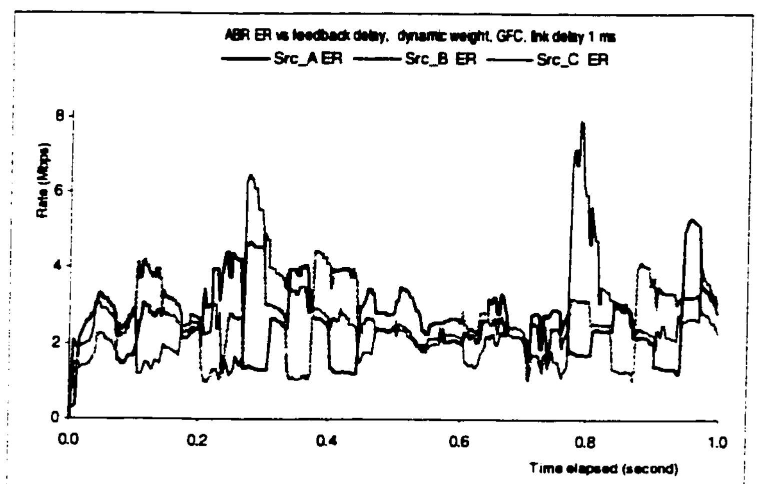


Figure 70 ER vs. delay, GFC, dynamic weight, d=1ms

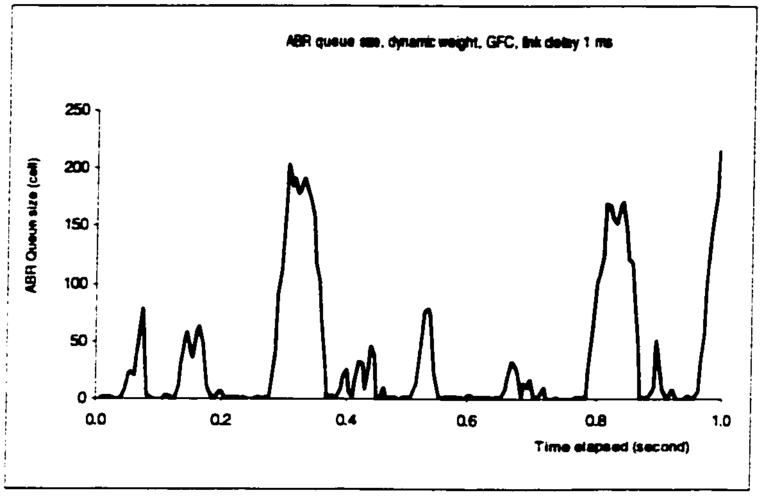


Figure 71 ABR queue size, GFC, dynamic weight, d=1ms

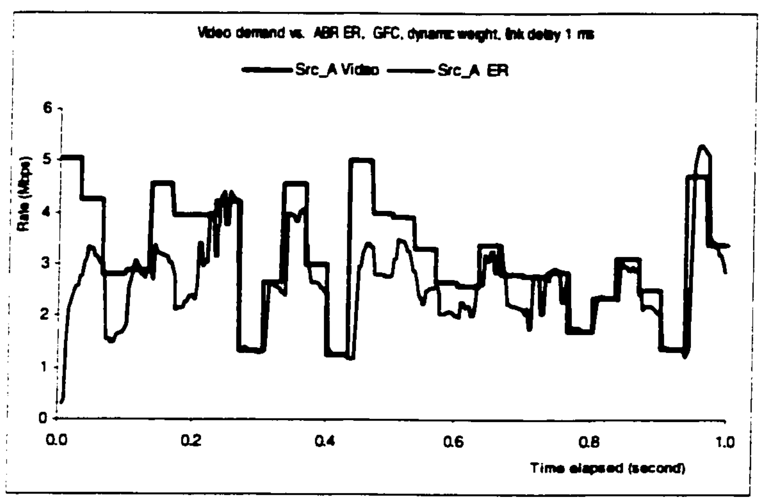


Figure 72 Src_A ER vs. demand, GFC, dynamic weight, d=1ms

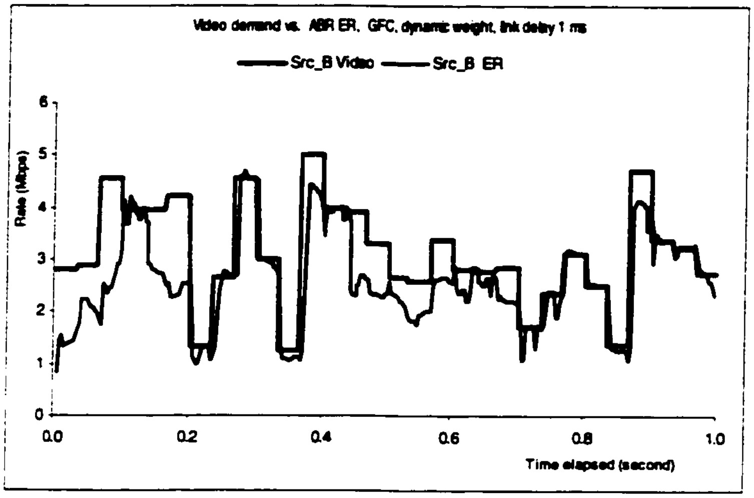


Figure 73 Src_B ER vs. demand, GFC, dynamic weight, d=1ms

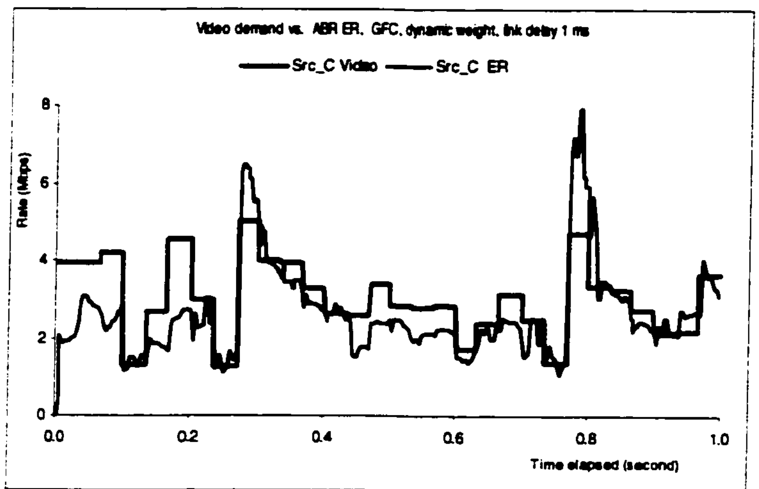


Figure 74 Src_C ER vs. demand, GFC, dynamic weight, d=1ms

The simulation results and analysis with the Generic configuration show that with long link delay (hence long larger control delay and feedback delay), neither fixed weight allocation nor dynamic scheme works well. With long feedback delay, the dynamic scheme makes the performance even worse due to the additional error of weight introduced by the long FRM travel time. With shorter propagation delay between the ABR sources and the bottleneck switch, both fixed weight allocation and dynamic weight scheme work well, though there is some impact due to the small delay. With dynamic weight allocation scheme, the total available bandwidth is allocated among the video ABR sources according to their demands dynamically. It improves the overall QoS of all the ABR sources by proportionally dropping the extra video data among all the ABR sources.

CHAPTER 5 CONCLUSIONS

In the first part of the thesis, we have studied the behavior of the FASTRAC rate control algorithm and the effect of a set of parameters and factors. The simulation results show that the FASTRAC rate control algorithm is simple and scalable to the number of connections and network configurations. It achieves fairness and stability when the control gain index meets certain condition which takes into account control delay, measurement update interval and link capacity. While the algorithm guarantees stability in general, a set of parameters have significant effect on the responsiveness and performance of the system. We summarize as follows:

- The RM cell interval has a considerable effect on the responsiveness of the system. In general, the smaller the RM cell interval, the better the responsiveness to the changing of the network resources. But the effect of RM cell interval also depends on the measurement update interval and the ABR rate. The selection of RM cell interval should consider not only system responsiveness but also management overhead. When the time between RM cells is much smaller than the measurement update interval, there is very little improvement to the system responsiveness to increase the RM cell frequency. A smaller N_{rm} value can still achieve good responsiveness when the ABR rate is high.

- The control gain factor α has a significant effect on the responsiveness and performance of the system. In general, the larger the control gain factor , the faster it converges but with more oscillation. A smaller control gain value takes longer to converge to the new target but has less rate oscillation. For system with long control delay, a smaller control gain can achieve better responsiveness. While selecting a conservative control gain factor can always guarantee the stability and convergence of the ABR system, the tuning of control gain via the stability factor (f) can significantly improve the responsiveness and performance. A proper control gain can be selected in practice to achieve particular effect.
- Control delay has significant impact on the responsiveness and performance of the system. A large control delay makes the rate control algorithm performance worse. It can affect the link utilization and increase the queuing time of the ABR cells. This can result on a larger cell transfer delay and jitter which is sensitive for rt-video application.

The other part of the thesis work explores the use of ABR to support real-time video applications. In order to achieve our objectives and get a comprehensive set of results, we have developed realistic traffic source models. By transmitting these realistic real-time video statistics with the FASTRAC simulator, we have proved that the FASTRAC rate control algorithm can support real-time video traffic over ABR service under certain scenarios. We also proposed the dynamic weight scheme in order to improve the performance of the FASTARC when transmitting real-time video traffic. By adding greedy traffic applications we can further improve the link utilization without affecting the real-time video transmission over ABR service. Therefore we summarize our simulation study with following conclusions:

- The FASTRAC rate control algorithm can support real-time video applications as well as greedy traffic over ABR connections when the control delay is small;
- By introducing dynamic weight allocation scheme, the performance of transmitting real-time video application over ABR connections is further improved, leading to the improvement of the QoS of the transmitted video application as well as the improvement of link utilization;
- The unused link budget can be further utilized by adding non real-time traffic applications, such as file transfer and email exchange, which further increase the overall link utilization without affecting the QoS of the rt-video applications;
- With long control delay connections, the performance becomes poor due to the large forward delay and the nature of rt-video application. With long delay to the bottleneck switch, the dynamic weight scheme introduces extra error to the fair share allocation and thus makes the performance even worse than the fixed weight scheme.

Due to the time limit of the thesis, in our study we only consider the supporting of un-partitioned MPEG-2 video traffic over ABR service. It would be interesting to further the study by exploring the performance of the system when supporting partitioned video traffic and smoothed video traffic. With partitioned video, when there is not enough bandwidth for the ABR source, the enhancement layers can be dropped. With smoothed video traffic that has less rate changes, the dynamic weight scheme may not be vulnerable to the long forward delay. Another direction for future work is to study the performance of the FASTRAC rate control algorithm when applying filtering or smoothing computation to the measurements of the VBR background traffic. Using filtering, such as moving average, the fair share computed for the ABR source will have less changes, making it easier to adapt by the ABR applications. But the control error may cause larger queuing delay which needs to be considered in the future study. Another way to reduce the impact of long propagation delay is to use virtual source and virtual destination (VS/VD) specified in the ATM Traffic Management. These can be the directions of the future research.

REFERENCES

1. Traffic Management Specification Version 4.1, ATM Forum/AF-TM-0121.000, March 1999. http://www.atmforum.org/pages/libraryfs1.html.

2. R. Goyal et al., "Modeling Traffic management in ATM networking with OPNet," Proc. of OPNETWORK'98, Washington DC, May 1998, http://www.cis.ohio-state.edu/~jain/papers/opnetwork98.

3. B. J. Vickers et al., "Feedback Control Mechanism for Real-Time Multipoint Video Services", IEEE Journal on Selected Areas in Communications, Vol.15, No.3, April, 1997, pp.512~530.

4. R. Jain, "Real_Time ABR, MPEG2 Streams over VBR, and Virtual Source/Virtual Destination". http://www.cis.ohio-state.edu/~jain/papers/.

5. R. Jain, S. Kalyanaraman et al., "Real-Time ABR: Proposal for a New Work Item", ATM Forum/96-1760, December 1996.

6. N. K. Agarwal, A. K. Gupta, W. Bin, L. Orozco Barbosa, "Scaleable Delivery Mechanism for VBR MPEG-2 Video Streams over the Internet", Proc. IEEE Pacific RIM Conference, Victoria, August 1999, 284-287.

7. U. Horn, B. Girord, "Scalable Video Transmission for the Internet", Computer Networks and ISDN Systems, Vol. 29, 1997, pp. 1833-1842.

8. B. Vandalore et al., "A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia", Journal of Real Time Systems, April 1999.

9. A. Friday et al., "Developing Adaptive Applications: The MOST Experience".

10. T. M. Chen et al., "The Available Bit rate Service for data in ATM Networks", IEEE Communications Magazine, Vol. 34, Issue 5, May 1996 pp: 56-58, 63-71

11. K. W. Fendick, "Evolution of Controls for the Available Bit Rate Service", IEEE Communications Magazine, Volume 34, Issue 11, Nov. 1996, pp 35~39.

12. R. Jain, S. Kalyanaraman et al., "ERICA Switch Algorithm: A Complete Description", ATM Forum/96-1172, August 1996.

13. B. Vandalore et al., "General Weighted Fairness and its Support in Explicit rate Switch Algorithms", Computer Communications Journal, February 1999.

14. S. Fahmy et al., "On Determining the Fair bandwidth Share for ABR Connections in ATM Networks", Proc. of ICC'98, Atlanta, June 1998, pp . 1485~1491.

15. K. B. Younes, K. Begain, "Scalable Video on Demand on ABR in ATM Networks", INFOCOM'97.

16. T.V. Lakshman et al., "Transporting Compressed Video over ATM Networks with Explicit Rate Feedback Control", INFOCOM'97, Proceedings, pp. 38~47.

17. N.G. Duffield, K.K. Ramakrishnan, A.R. Reibman, "SAVE: An Algorithm for Smoothed Adaptive Video over Explicit Rate Networks", Proc. INFOCOM'98, San Francisco, March-April 1998, pp. 1093-1102.

18. R. Goyal et al., "Effect of RM cell Interval on ABR Feedback: A Simulation Study using OPNet", http://www.cis.ohio-state.edu/~jain/papers/.

19. P. Cuenca, A. Garrido, F. Quiles, L. Orozco-Barbosa, "An Efficient Protocol Architecture for Error-resilient MPEG-2 Video Communications over ATM Networks", IEEE Trans. on Broadcasting, Vol.45, No.1, March 1999.

20. P. Cuenca, L. Orozco-Barbosa, F. J. Quiles, and A. Garrido, "Loss-Resilient ATM Protocol Architecture for MPEG-2 Video Communications", IEEE Journal on Selected Areas in Communications, Vol. 18 , No. 6, June 2000.

21. B. Vandalore et al., "QoS and Multipoint Support for Multimedia Applications over the ATM ABR service", IEEE Communications Magazine, January 1999.

22. W. Feng et al., "Performance Evaluation of Smoothing Algorithms for Transmitting Prerecorded Variable-Bit-Rate Video", IEEE Trans. on Multimedia, Vol.1, No.3, September 1999, pp.302~312.

23. J.Aweya, M.Ouellette, D.Y.Montuno, "A Simple, Scalable, Provably Stable, Explicit rate Computation Scheme for Flow Control in Computer Networks", Int'l Journal of Communication Systems.

24. F. Bonomi, K. W. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service", IEEE Network, March/April 1995, pp.25~39.

25. J. P. Chew, A. K. Gupta, L. Orozco-Barbosa, "Improving Fairness of Long VCs with Dynamic Weights in the ATM ABR Service", Proc. 1999 SPIE Performance and Control Conference, Boston, U.S.A, September 1999, pp. 150-160.

26. R. Jain et al., "Source Behavior for ATM ABR traffic management: An Explanation", IEEE Communications Magazine, November 1996.

27. V. Bhaskaran and K. Konstantinides, Image and Video Compression Standards. Algorithms and Architectures, Second edition, Kluwer Academic Publishers, 1997.

28. G. Lu, "Communication and Computing for Distributed Multimedia Systems", 1996 Artech House.

29. E. Moyano, P.J. Garcia, A. Garrido, F.J. Quiles, L. Orozco-Barbosa. J. Duato, "Parallel Wavelet Compression System on Myrinet for Medical Videos", First Myrinet Users Group Workshop, Lyon, September 2000.

30. P.N. Topiwala. Wavelet image and video compression, Kluwer Academic Publishers, 1998.

31. A. Campbell, G. Coulson et al., "Supporting Adaptive Flows in a Quality of Service Architecture", ACM Multimedia Systems Journal, 1996.

32. J. Zhang, J. Hui, "Applying traffic smoothing techniques for quality of service control in VBR video transmission", Comput. Commun., Vol. 21, pp. 375-389, Apr. 1998.

33. H. Zhang, E. W. Knightly, "A New Approach to Support Delay-Sensitive VBR Video in Packet-Switched Networks", Proc. 5th Workshop on Networking and Operating System Support for Digital Audio and Video, pp. 275-286, Apr. 1995.

34. OPNET Modeling Manual Vol 1, NIL 3 Inc.,