Studies of Dynamic Bandwidth Allocation for Real-Time VBR Video Applications

Mei Han

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Committee:
Dr. Yao Liang, Committee Chair
Dr. Luiz A. DaSilva, Committee Member
Dr. Yue Wang, Committee Member

Feb 23, 2005
Alexandria, Virginia

Keywords: Dynamic network bandwidth allocation, Traffic model, Traffic Prediction

© Copyright 2005, Mei Han

Studies of Dynamic Bandwidth Allocation for Real-Time VBR Video Applications

Mei Han

(Abstract)

Variable bit rate (VBR) compressed video traffic, such as live video news, is expected to account for a large portion of traffic in future integrated networks. This real-time video traffic has strict delay and loss requirements, and exhibits burstiness over multiple time scales, thus imposing a challenge on network resource allocation and management. The renegotiated VBR (R-VBR) scheme, dynamically allocating resources to capture the burstiness of VBR traffic, substantially increases network utilization while satisfying any desired quality of service (QoS) requirements. This thesis focuses on the performance evaluation of R-VBR in the context of different R-VBR approaches. The renegotiated deterministic VBR (RED-VBR) scheme, proposed in [5], is thoroughly investigated in this research using a variety of real-world videos, with both high quality and low quality. A new Virtual-Queue-Based RED-VBR is then developed to reduce the implementation complexity of RED-VBR. Simulation results show that this approach obtains a comparable network performance as RED-VBR: relatively high network utilization and a very low drop rate. A Prediction-Based R-VBR based on a multiresolution learning neural network traffic predictor [10] is studied and the use of binary exponential backoff (BEB) algorithm is introduced to efficiently decrease the renegotiation frequency. Compared with RED-VBR, Prediction-Based R-VBR obtains significantly improved network utilization at a little expense of the drop rate. This work provides evaluations of the advantages and disadvantages of several R-VBR approaches, and thus provides a clearer big picture on the performance of the studied R-VBR approaches, which can be used as the basis to choose an appropriate R-VBR scheme to optimize network utilization while enabling QoS for the application tasks.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my academic and research advisor Dr. Yao Liang for his inspiration and constant encouragement during my graduate studies. He was very patient and always provided his excellent guidance in helping me to conduct and complete this research work. I would like to convey my sincere thanks to Dr. Luiz A. DaSilva for his expertise and his gracious service in reviewing this work. I am truly grateful to his efforts to improve my technical writing skills. I would also like to thank Dr. Yue Wang for serving on my advisory committee and presenting good advices on some topics in this work.

I am indebted to Dr. Pamela Kurstedt and Dr. Saifur Rahman for their generous support. The benefit I received as a graduate assistant made my graduate studies possible.

I would like to thank the people I have come to know at Virginia Tech in Blacksburg campus and in Northern Virginia campus, whose friendship I will always enjoy. Finally, I want to extend my profound appreciation to my beloved parents and my husband for their love, affection, and invaluable support over years.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

High speed packet-switching networks are expected to carry a variety of applications with a wide range of quality of service (QoS) requirements. Multimedia applications, such as IPTV and online news, are amongst the list of future applications with significant growth potential. For example, the tendency that more and more hardware vendors are now upgrading the home PC to a multimedia center to play music and video has heightened the need for efficiently and reliably transmitting diverse multimedia information through packet-switching networks. Most multimedia applications, such as VBR compressed video, require a network resource reservation mechanism to guarantee their strict delay and loss requirements. On the other hand, due to their bursty nature, it is difficult for network resource allocation and management (NRAM) to efficiently allocate network resources to achieve high overall network utilization. Desired QoS for individual applications and high overall network utilization are two conflicting design goals in the realm of network communications, posing a challenge in the design of NRAM systems. In recent years, researchers have studied and proposed different NRAM schemes. Among them, the renegotiated VBR (R-VBR) scheme for VBR compressed video applications seems to achieve good performance due to the possibility of keeping the overall network utilization high while fulfilling any desired QoS requirements. This thesis work attempts to further explore R-VBR through a simulation-oriented approach, provides our observation and analysis on R-VBR, and presents new methods to improve the current R-VBR scheme.

Section 1.1 of this chapter gives a literature review of different NRAM schemes and provides the background information needed in this research. Section 1.2 outlines our research focus and research objectives. Finally, section 1.3 describes the organization of this thesis.

1.1 Background

1.1.1 Characteristics of VBR Compressed Video

Recent studies (e.g., [1] - [4]) have shown that VBR compressed video usually exhibits bit rate variations over multiple time scales.

Short-time-scale variations, also called short-term burstiness, are primarily due to the coding scheme. MPEG encoded video streams consist of three types of frames, intracoded frame (I frame), forward predicted frame (P frame) and bidirectional predicted frame (B frame). P frame and B frame are used to reduce temporal as well as spatial redundancies by employing inter-frame compression while I frame is compressed to reduce only the spatial redundancies present in an image sequence. Intercoded frames (P frame and B frame) therefore require fewer bits than intracoded frames (I frame). The I, B, P frames are alternately arranged in a periodic pattern. The pattern is formally called a Group of Picture (GOP). Such a compression structure can greatly increase the efficiency of data compression, but also introduce short-term burstiness, usually persisting in a relatively short time like one or several GOPs. Figure 1.1.1.1 illustrates the frame distribution of a 5.6-second long segment of a video sequence of the *Star Wars IV* movie with high quality encoding, which is encoded at the frame rate of 25 frames / second. It can be seen from the plot that its GOP pattern is IBBPBBPBBPBB. I frames have much larger frame size than B frames and P frames. The brief spikes caused by I frames appear periodically every 0.48 s, that is the duration of one GOP.

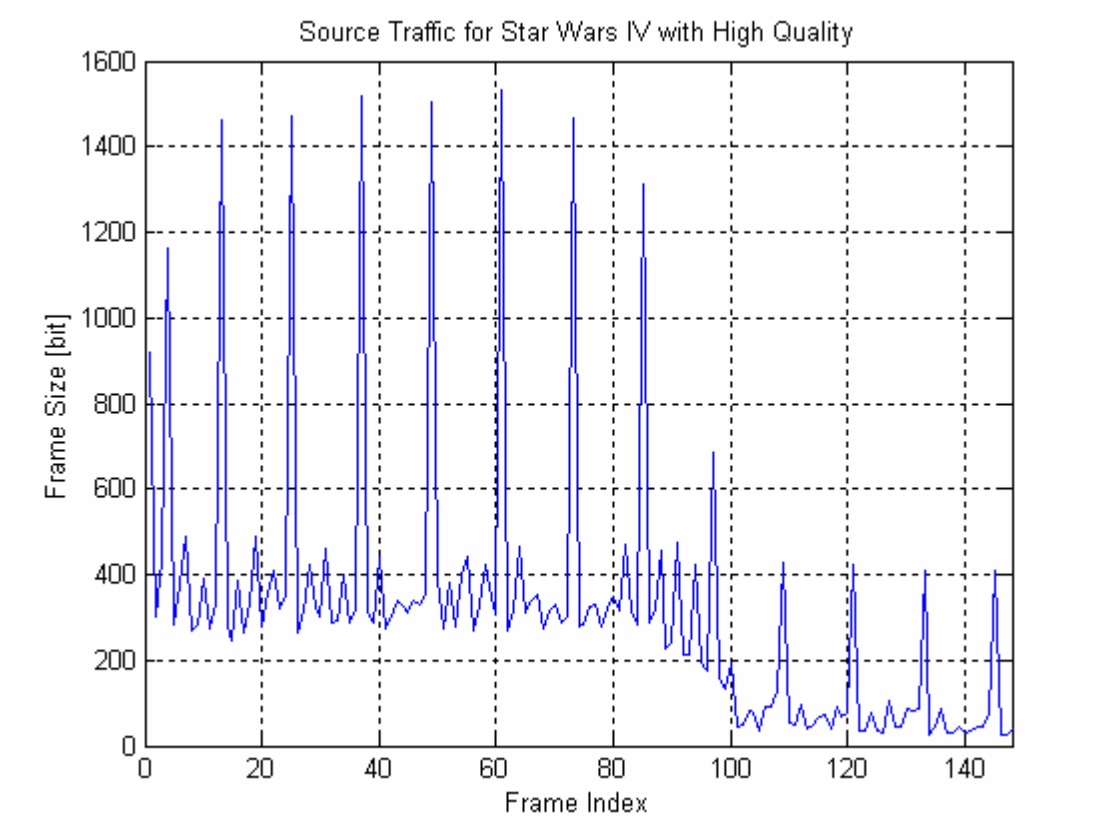


Figure 1.1.1.1: Distribution of a 5.6-second long MPEG-4 video trace

Long-time-scale variations, also called long-term burstiness, correspond to the change of the information content of different shots and scenes. Bursts of this type may persist for a relatively long time normally in a range of tens of seconds to minutes. Figure 1.1.1.2 shows the frame distribution of an entire video sequence of the *Star Wars IV* movie with high quality, from which we can easily identify that the frame sizes stay roughly at a relatively fixed level during long periods. The variation of frame length between the periods is generally due to a scene change.

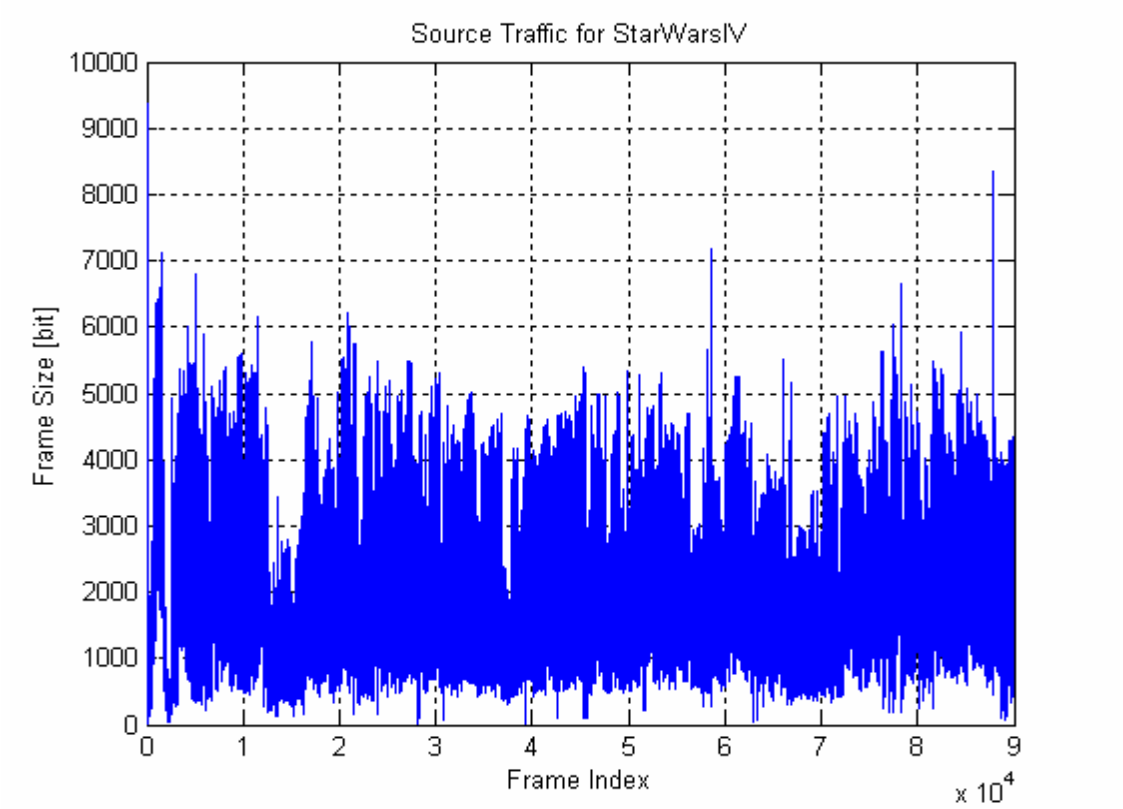


Figure 1.1.1.2: Distribution of an entire MPEG-4 vide trace

Since the long-term burstiness generally persists for a relatively long period, it has an important influence on the overall network performance. Suppose that each video stream transmits at a rate lower than the peak rate of the entire video stream. When the overall rate of the aggregated sources is greater than the link speed, the bursts begin to fill up the buffer in the end system. When this situation lasts for a while, the buffer capacity should be large enough to accommodate the aggregated bursts. Otherwise, the loss rate may be unexpectedly high. The large buffer capacity corresponds to a long delay for the sources. The long delay or the large loss rate would greatly degrade the QoS of the applications, which cannot be tolerated by some VBR compressed applications, such as on-line applications and interactive applications. Consequently, VBR compressed video requires a guaranteed QoS during transmission. A common approach to provide the desired QoS to individual sources is to implement a network resource reservation mechanism for each stream and to police the admitted connection based on its required QoS. However, the bursty nature of VBR compressed video complicates the design of NRAM systems. On

the one hand, VBR compressed video requires a guaranteed QoS during transmission and this can be done by allocating a bandwidth close to its peak rate for the entire video stream; but on the other hand, the objective of the NRAM design is to efficiently utilize network resources, that is, to achieve high utilization. Since the peak rate of VBR compressed video stream is significantly greater than the average rate, the peak-rate allocation during the connection life time would be wasteful of network bandwidth resources. In order to support individual video streams with different desired QoS requirements while keeping the overall high network utilization, NRAM systems should be carefully designed to explore reasonable and potential statistical multiplexing gain (SMG) among VBR compressed video streams.

1.1.2 The Place to Employ NRAM

Networks can be logically divided into two primary parts, backbone networks and edge networks [6, 7]. The edge networks typically consist of low capacity switches and links. They serve as local area networks (LAN) as well as the access networks (AN) to the backbone networks for local clients. The backbone networks, comprising powerful and high bandwidth switches/routers and links, provide communication facilities to the aggregated traffic coming from the edge networks. The considerable capacity of the backbone networks makes it possible to simultaneously process many more connections than the edge networks. Therefore, the bottleneck of the network resources due to the increasingly aggregated traffic more typically occurs in the edge networks [7].

The NRAM system is generally implemented in the edge networks due to two reasons [8]. First, this will greatly conserve the computational resources in the backbone networks. Secondly, the NRAM system, set nearby the traffic sources, can easily and directly extract traffic statistics and QoS requirements from individual sources and then immediately make bandwidth allocations in response to their significant bit rate variations.

1.1.3 Literature Review

Due to the unique characteristics of VBR encoded video, it is difficult to obtain very high network utilization while providing the guaranteed QoS to each VBR compressed video source. Many research works, presenting different NRAM schemes, deal with an optimal tradeoff between network utilization and QoS. These schemes may be categorized into several types, building on the work in [9]. We illustrate these types in figure 1.1.3.1.

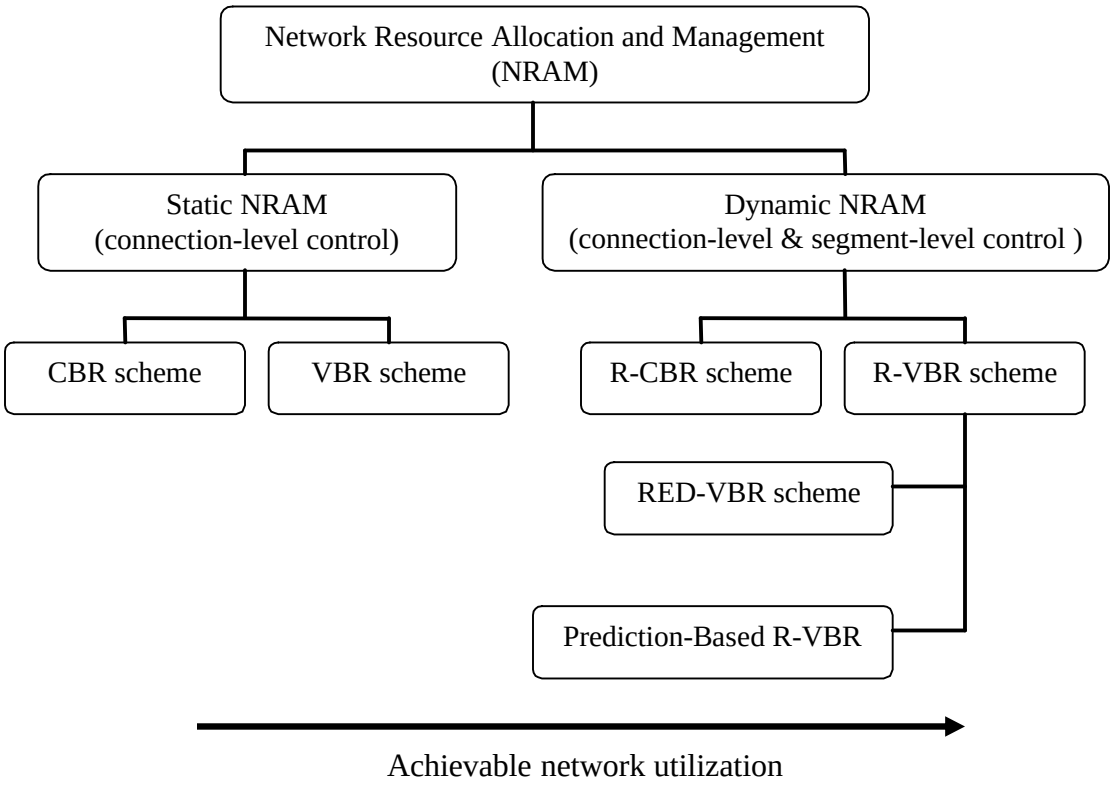


Figure 1.1.3.1: NRAM design structure

According to the level of the management control at which the NRAM system works, the NRAM system may be generally classified as static NRAM and dynamic NRAM.

Static NRAM employs a connection-level control mechanism, exclusively allocating network resources to each video source only at the time of its connection establishment. Consequently, static NRAM has no multiplexing function. The most popular schemes used in the static NRAM system are static constant bit rate (CBR) and static VBR. In static CBR, a source is restricted to a fixed bit rate during the lifetime of the connection. Such an allocation scheme has some drawbacks when it applies to VBR compressed video applications. For example, the peak rate allocation scheme, allocating network resources in terms of the source's peak rate, provides deterministic guaranteed service to the video sources but achieves very low network utilization. In order to increase the network utilization, static CBR generally chooses a bit rate lower than the peak rate, like the long-term average rate. During the period when the peak rate is much greater than the long-term average rate, packet loss inevitably occurs. Static VBR addresses this issue to some degree. One approach in static VBR is to choose a token rate in conjunction with a large token bucket to transport VBR compressed video streams and accommodate part or all of the sustained bursts [9]. The other approach, called the deterministic VBR (D-VBR) scheme [5], identifies a set of bounding rates over multiple interval lengths to explicitly describe the behavior of burstiness of VBR compressed traffic in a rough way. This approach provides a deterministic service guarantee. It is worth noting that both approaches use more than one rate to implement resource allocation and are capable of exploring bit rate variations among different VBR traffic sources. Static VBR, therefore, can obtain higher network utilization than static CBR for the same level of QoS. However, the static NRAM system, employing either static CBR or static VBR, has the following intuitive limitations.

1. The source gets the allocated bandwidth only at the time of the connection establishment. If the reserved bandwidth is not appropriate, either the source will suffer from a large delay or the utilization will be low.
2. Static NRAM is suitable for pre-recorded VBR compressed video applications, but not for real-time VBR compressed video applications. This is because for the pre-recorded VBR compressed video applications, static NRAM is able to extract the entire traffic information of a video stream stored in a remote multimedia

server before its transmitting, and thus make the appropriate network resource allocation. For the real-time VBR compressed video applications, the entire traffic information cannot be obtained in advance, which results in an inaccurate estimation of the required network resources.

Real-time applications absolutely need online processing instead of a one-time bandwidth allocation at the time of connection establishment. The dynamic NRAM system best addresses the shortcomings of the static NRAM system. In addition to connection-level control, dynamic NRAM also performs control at a segment level of the video stream, i.e., the burst layer. Each source is transmitted according to its currently reserved resources and periodically renegotiates its resources with the dynamic NRAM system over time. When the required resources, refreshed on the fly, change significantly, in the sense of either violating or dropping significantly below the previously reserved resources, a new request is dynamically initiated. The request can be rejected or admitted according to the available network resources. High SMG is accordingly achieved because the unused resources released by some sources can be efficiently utilized by some other sources. Correspondingly, the increase of the SMG leads to the increase of the network utilization. Here, the time point to generate a resource renegotiation request is defined as a ‘renegotiation point’ and the video sequence between two adjacent renegotiation points is labeled as a ‘segment.’ Network resources are reserved on a per segment basis. Since the renegotiation request requiring additional network resources may be denied/blocked due to the network resource limitations, the dynamic NRAM system introduces a segment-level blocking probability to describe the probability of renegotiation failure.

There are two important issues in the design of the dynamic NRAM system. 1) When to renegotiate? 2) How much resources to ask for? These two issues affect the performance metrics, such as the renegotiation frequency and the network utilization. The renegotiation frequency is an important performance metric in the dynamic NRAM system. A high renegotiation frequency leads to high SMG, but introduces excessive control overhead to the network. The overall network performance, therefore, should take into account the tradeoff between the renegotiation frequency and the network utilization.

Of all the proposed schemes in the dynamic NRAM system, the determination of renegotiation points generally falls into one of three categories [8]: deterministic renegotiation points, which are empirically set according to a fixed renegotiation interval; traffic-based renegotiation points dynamically generated in response to bit rate variations of VBR compressed video traffic; and content-based renegotiation points, which occur due to a change in scene contents.

Generally, two types of schemes in the category of the dynamic NRAM system are available, renegotiated CBR (R-CBR) and renegotiated VBR (R-VBR). R-CBR [9] and R-VBR add the renegotiation mechanism to static CBR and static VBR, respectively. Both of them try to extract the potential SMG at the segment level to obtain the achievable high network utilization. Since R-VBR employs static VBR at the segment level, it may be able to obtain more SMG than R-CBR.

The efficiency of R-VBR depends on several aspects, such as the renegotiation frequency, the selection of the appropriate traffic descriptor and the accuracy of the traffic model. Many studies have been proposed toward dealing with each of these aspects. Zhang *et al.*'s research presents the renegotiated deterministic VBR service (RED-VBR) [5]. This scheme provides a deterministic service on the segment level in conjunction with a statistical service on the connection level to extract the available SMG, and performs a graceful adaptation in response to a successful renegotiation during the connection period. Such a design idea has been widely referred by other studies in this area and has been generally chosen as a baseline to make performance comparisons. Note that RED-VBR proposed in [5] did not make use of a traffic predictor. Instead, it directly utilizes the traffic parameters of the currently extracted frames to make a bandwidth reservation for the incoming frames. There have been several other investigations into R-VBR, such as Bocheck *et al.* [7] and Wu *et al.* [8]. Both approaches focused on developing an efficient traffic predictor and investigating the impact of traffic prediction on performance. We call the R-VBR scheme integrated with a traffic predictor the Prediction-Based R-VBR scheme. Compared with RED-VBR, Prediction-Based R-VBR first predicts the sizes of the incoming frames, and then issues a renegotiation

request for the incoming frames whenever the predicted traffic characteristic is substantially inconsistent with that for which the current bandwidth is allocated before. Figure 1.1.3.2 illustrates this procedure in the dashed arrows.

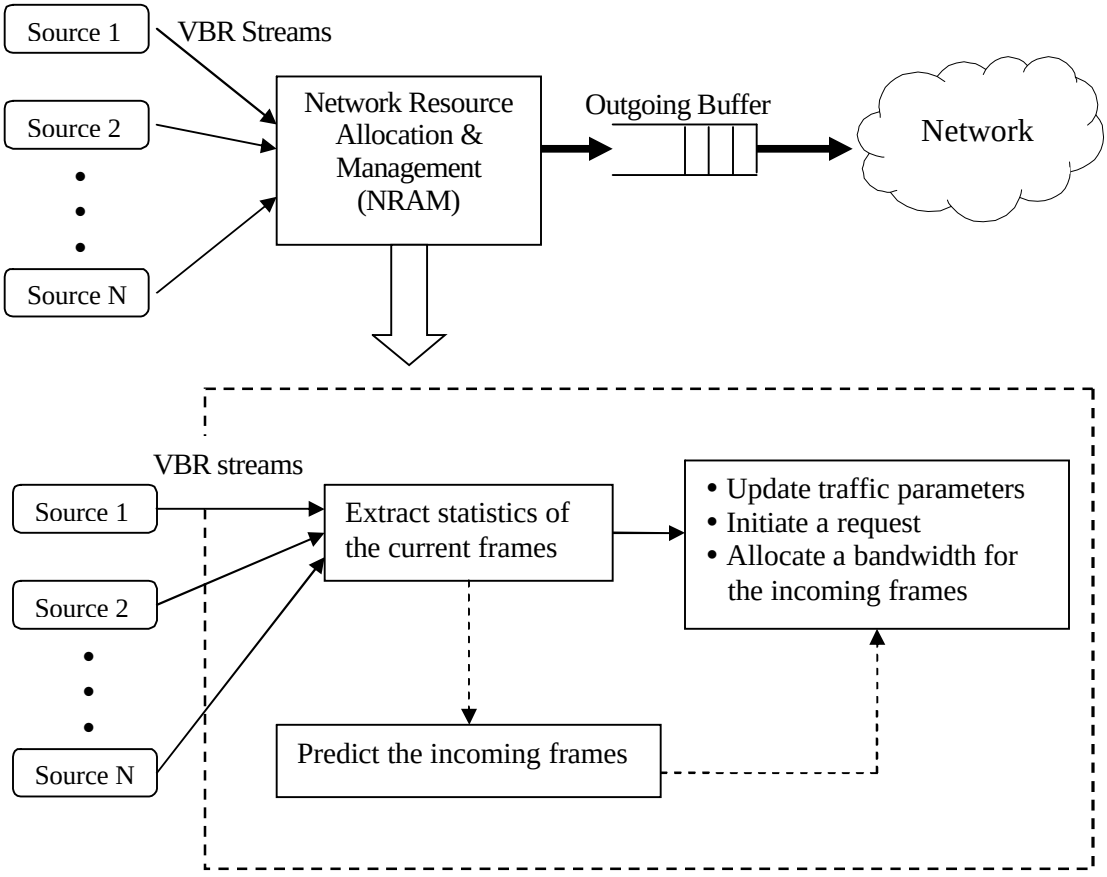


Figure 1.1.3.2: Comparison between RED-VBR and Prediction-Based R-VBR

1.2 Research Motivations and Objectives

Many studies of R-VBR, described in section 1.1.3, have been devoted to present a better solution to address the following issues: determining the renegotiation frequency, choosing an appropriate traffic model and developing an accurate traffic predictor. The published studies [5, 7, 8], based on our best knowledge, evaluated the proposed scheme based on simulations of one or two MPEG-1 or MPEG-2 video traces. We think that substantially more experiments on a variety of video traces would be very helpful in rigorously investigating and evaluating the advantages and disadvantages of any R-VBR scheme. In this research, we first study RED-VBR proposed in [5] in detail, and then develop a novel Virtual-Queue-Based RED-VBR scheme to significantly simplify the implementation of RED-VBR. The proposed Virtual-Queue-Based RED-VBR is evaluated on a wide variety of real-world MPEG-4 VBR compressed video traces [16] and is compared with RED-VBR and its variant. In addition, we also investigate a new Prediction-Based R-VBR scheme where a novel network traffic predictor developed by Y. Liang [10] is employed.

The objectives of this research include:

- To propose a Virtual-Queue-Based RED-VBR scheme to simplify the deployment of RED-VBR as well as the tuning of network parameters.
- To test Virtual-Queue-Based RED-VBR and RED-VBR on various real-world MPEG-4 encoded video traces, evaluate and compare their performance.
- To develop a Prediction-Based R-VBR scheme by integrating a traffic predictor, propose that the use of the binary exponential backoff (BEB) algorithm can decrease the renegotiation frequency, and examine the performance of Prediction-Based R-VBR.
- To conduct an analysis on the advantages and disadvantages of existing and new R-VBR approaches in terms of QoS requirements, network utilization and

tradeoffs, and provide a clearer big picture on the performance of several R-VBR approaches.

1.3 Thesis Outline

The remainder of this thesis is organized as follows. In chapter 2, a new Virtual-Queue-Based RED-VBR algorithm is introduced to reduce the computation and implementation complexity of RED-VBR. We then evaluate and compare the performance of both algorithms based on a variety of video traces both with high quality and with low quality. Based on simulation results, the tradeoff among different performance metrics is discussed for various scenarios. Chapter 3 develops a Prediction-Based R-VBR scheme based on a novel network traffic predictor. We evaluate our Prediction-Based R-VBR, compare it with RED-VBR and discuss their performance. Finally, Chapter 4 concludes the thesis by summarizing our contributions and providing some thoughts about further research in this area.

Chapter 2

R-VBR: RED-VBR

Chapter 1 provided the necessary background needed to understand this thesis, including an introduction of characteristics of VBR compressed traffic and a literature review of different schemes in NRAM systems. This chapter first gives a brief overview of the deterministic bounding interval-length dependent (D-BIND) traffic model [11] in section 2.1, which is used in both RED-VBR and Prediction-Based R-VBR presented in chapter 3. RED-VBR introduced in [5] is then described in section 2.2. Section 2.3 presents some important issues observed from experiments with RED-VBR, which were not clearly indicated in [5]. The Virtual-Queue-Based RED-VBR scheme, a new solution to address those issues, is proposed in section 2.4. Section 2.5 outlines the simulation set up. Our observation and analysis of simulation results are included in section 2.6. Finally, the main ideas presented in this chapter are summarized in section 2.7.

2.1 Overview of D-BIND Traffic Model

A traffic model, also called traffic descriptor, describes the characteristics of a certain traffic with the use of countable traffic parameters. Several traffic models have been proposed in recent years. One popular traffic model is the deterministic bounding interval-length dependent (D-BIND) traffic model [11]. It has been widely recognized

and utilized in NRAM systems for multimedia applications due to the following advantages:

1. The traffic model specifies the traffic by a set of bounding rates over several different intervals to accurately capture the burstiness of VBR compressed video.
2. It makes admission control and policing easy to be implemented because of its countable parameters, that is, rate-interval pairs.
3. The traffic model helps smooth out part of the bursts because the D-BIND bounding rate is essentially the maximum allowed arrival rate for the associated interval.
4. It can be demonstrated that, compared with static CBR, static VBR can provide a deterministic service as well even in the worst case when it makes use of the D-BIND traffic model, and achieve a substantial improvement in network utilization [11].

Therefore, we choose the D-BIND traffic model to investigate the behavior of R-VBR in our research.

Denote by $A[\tau, \tau + t]$ the cumulative number of bits generated by a source arriving in the interval ranging from τ to $\tau + t$, and the empirical envelop $B^*(t)$, the tightest time-invariant bound over the interval, is:

$$B^*(t) = \sup_{\tau > 0} A[\tau, \tau + t], \ \forall t > 0 \qquad (2.1\text{-}1)$$

The D-BIND traffic model [11] characterizes the arrival traffic with a set of rate-interval pairs $R_T = \{(r_k, t_k) \mid k = 1,2,..., P\}$, where $r_k = q_k / t_k$. Here, r_k is the bounding rate and q_k is the total number of bits over the interval t_k. Therefore, the traffic constraint function B_{R_T} of the D-BIND traffic model is:

$$B_{R_T}(t) = r_k t_k + \frac{r_k t_k - r_{k-1} t_{k-1}}{t_k - t_{k-1}}(t - t_k),\ t_{k-1} \le t \le t_k \qquad (2.1\text{-}2)$$

with the assumption of $B_{R_T}(0) = 0$ at $t_0 = 0$.

Define one frame time to be a time unit (note that this simplification is applicable to the rest of the thesis work). For a video sequence with M frames, the tightest D-BIND traffic constraint function $B^*_{R_T}$ should be constructed by the consecutive P rate-interval pairs where P = M. It is a piecewise-linear time-invariant curve as shown in figure 2.1.1.

The relationship among the cumulative arrival function $A(0,t)$, the empirical envelope $B^*(t)$, the tightest D-BIND traffic constraint function $B^*_{R_T}(t)$ and the general D-BIND traffic constraint function $B_{R_T}(t)$ is illustrated in figure 2.1.1.

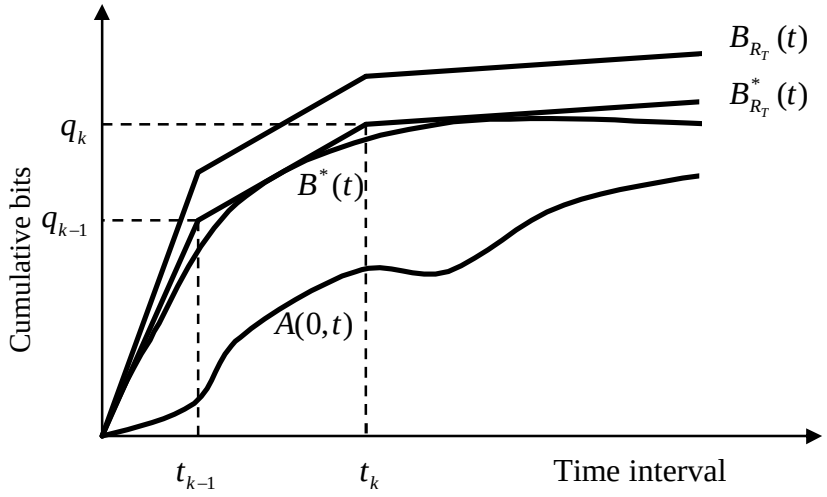


Figure 2.1.1: Comparison of traffic constraint functions

The following example depicts the application of the algorithm to compute the D-BIND rate-interval pairs. Assume that a video segment has 100 consecutive frames, which is defined to be a D-BIND window size (denoted by M). The frame rate of the video segment is 25 frames / second. For the interval length of $t_1 = 1/25\,s$, the bounding rate r_1 of this video segment can be determined by dividing the largest frame size of the 100

frames by t_1. Equivalently, when the interval t_{12} is $12 * 1/25\,s$, the bounding rate of r_{12} of the video segment can be computed by dividing the largest sum of any consecutive 12 frame sizes of the 100 frames by t_{12}. The D-BIND bounding rate is actually the maximum rate allowed for the associated interval. This allows smoothing out some level of bursts. The D-BIND traffic model always upper bounds the video segment in any selected interval length.

Figure 2.1.2 shows the D-BIND rate-interval pairs of video traces of the *Mr. Bean* movie and the *ARDTalk* talk show. There are totally 89,998 frames in both video traces, lasting about one hour. The total number of bounding rates P is chosen as 24 and interval length varies from one frame time to 24 frame times (1s) with a step size of one frame time.

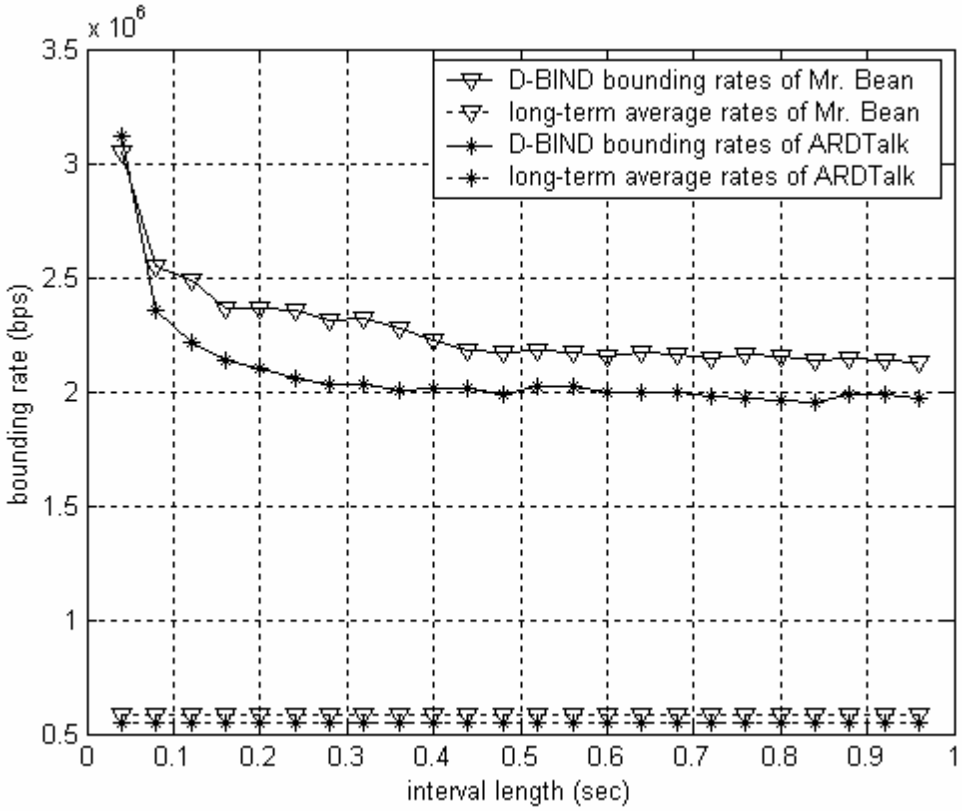


Figure 2.1.2: D-BIND rate-interval pairs

As shown in figure 2.1.2, when the interval length is one frame-time, the bounding rate of *Mr. Bean* is around 3.05 Mbps, which is the peak rate for the entire video sequence. The bounding rate is generated by the largest burst-size over the short-term of one frame time. When the interval length increases from one frame time to five frame times (around 0.2s), the bounding rate drops to 2.38 Mbps. After the interval length of 0.2s, the bounding rate slowly decreases. The same decreasing pattern can be found in the video trace of *ARDTalk*. Such a tendency reflects the short-term burstiness and the long-term burstiness of VBR compressed video traffic. In addition, the decrease of *Mr. Bean* from the peak rate to the long-term average rate is relatively slower than that of *ARDTalk*. This indicates that the *Mr. Bean* is more bursty than *ARDTalk*.

2.2 Overview of RED-VBR

The renegotiated deterministic variable bit rate (RED-VBR) scheme [5] uses the D-BIND traffic model to specify the traffic parameters, and implements a renegotiation mechanism to extract the achievable SMG during the workload periods. Therefore, for the same QoS requirement, RED-VBR would achieve higher network utilization than static CBR, static VBR and R-CBR.

2.2.1 Framework of RED-VBR

Figure 2.2.1.1 illustrates the framework of RED-VBR, similar to the framework in [7]. Note that this framework is generally applicable to other R-VBR schemes in the dynamic NRAM system as well.

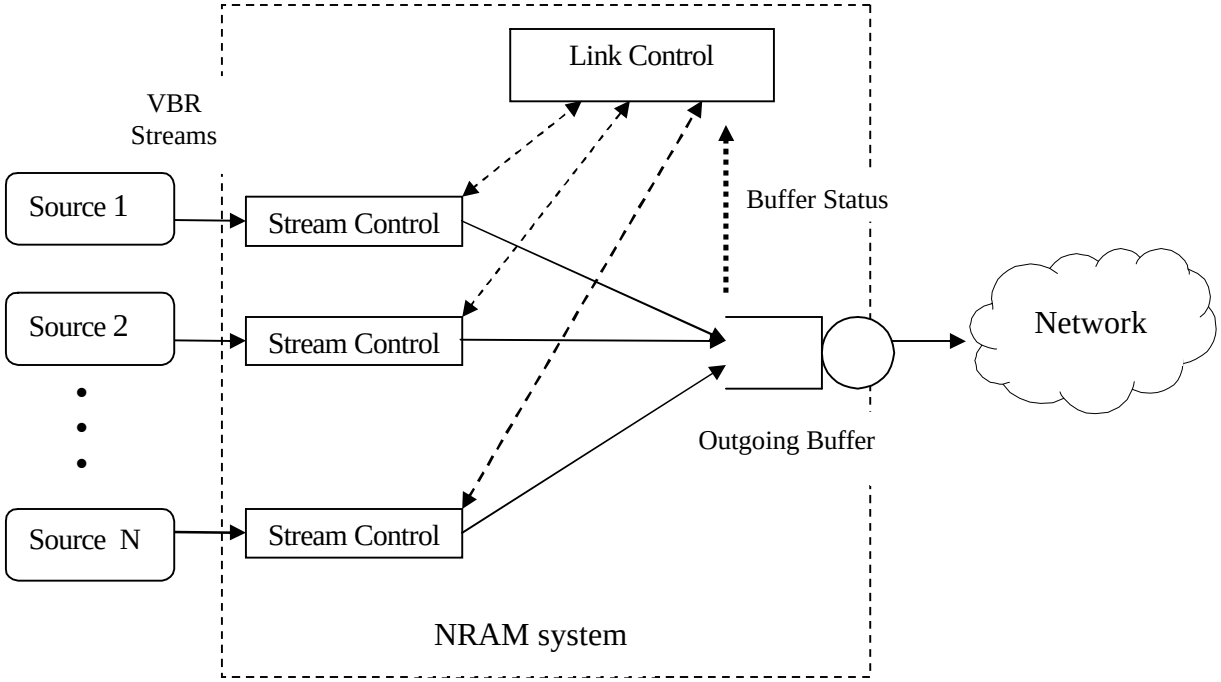


Figure 2.2.1.1: Framework of RED-VBR

It can be noted that in figure 2.2.1.1, the dynamic NRAM system with RED-VBR includes two primary function modules, stream control module and link control module. The stream control module has three key functions: 1) to extract the bit statistics of VBR encoded video stream from the corresponding source; 2) to calculate the traffic parameters according to the embedded traffic model; 3) to conduct the segmentation algorithm, determining when to renegotiate and how much bandwidth to request. The primary function of the link control module is to perform resource allocation according to the available resources. In general, the link control module is invoked by any request from a stream control module.

There are two primary procedures for RED-VBR.

1. Admission control procedure activated by connection requests.

   When a new source requests admission into the network, the stream control module sends a connection request to the link control module. Depending on the current link capacity, the bounding rates of the admitted ongoing connections and requirements of

bounding rates for the new connection, the link control module performs an admission control check to determine if this connection request may be admitted without degrading the QoS of other admitted connections.

2. Renegotiation control procedure motivated by renegotiation requests.

Once a source is admitted into the network, the stream control module begins to update the D-BIND bounding rates frame by frame using the currently loaded frames (in the case of RED-VBR) or a prediction regarding the incoming frames (in the case of our Prediction-Based R-VBR). At the same time, in response to a significant bit rate change, the segmentation algorithm periodically initiates a renegotiation request and immediately sends this request to the link control module. The link control module then conducts renegotiation control (similar to admission control) to determine to either accept or reject the renegotiation request. If a renegotiation request is rejected, the source would continue transmitting using the previously reserved resources by applying traffic shaping. Intuitively, the renegotiation mechanism divides the video sequence into multiple segments. During this dynamic procedure, the unused bandwidth released by some existing connections can be efficiently utilized by other admitted connections. High SMG can be consequently expected.

It is worth mentioning that with the use of the D-BIND traffic model, RED-VBR actually builds a statistical service on top of a deterministic service. The two types of services are built on different levels. The deterministic service is built on the segment level. Since each source has to transmit according to the reserved D-BIND bounding rates, a deterministic service, guaranteeing no packet dropped or delivered beyond its delay bound, is provided for each segment. On the connection level, RED-VBR provides a statistical service because the renegotiation request for a new segment may be denied or blocked due to the limited network resources. A renegotiation blocking probability is introduced here to describe the probability of renegotiation failure for RED-VBR.

2.2.2 Segmentation Algorithm

The goal of the segmentation algorithm is to determine when to renegotiate and how much more or less resources to request for the admitted connections. Intuitively, a higher renegotiation frequency may help extract more SMG, thus leading to higher network utilization. However, the higher frequency of renegotiations results in more network control overhead. It is expected to make a balance between renegotiation frequency and network utilization.

RED-VBR specifies two segmentation algorithms: offline and online [5]. Offline video segmentation algorithm can be applied to pre-recorded video streams. The algorithm takes an advantage of knowledge of the full video trace history to calculate the optimal renegotiation schedule in advance. Online video segmentation algorithm is used for real-time video streams like live news and video conferencing. An online algorithm uses the previous trace history to estimate future resource requirements and therefore generate a renegotiation request in advance. Online segmentation algorithms have generated great interest in recent years in the field of real-time VBR compressed video traffic. The focus of our research is on online real-time scenarios.

The online segmentation algorithm may be divided into two procedures: renegotiation requests for more resources and renegotiation requests for resource release. In general, the renegotiation request for resource release is always granted. For each admitted connection, the stream control module refreshes P measured bounding rates over time based on the previous M frames ($M \geq P$) and at the same time compares the measured bounding rates with the reserved bounding rates according to the segmentation algorithm to determine whether or not to issue a renegotiation request.

- Procedure for renegotiation requests for more resources

  Whenever any rate in the measured bounding rates $(r_1, r_2, ..., r_P)_{measured}$ is larger than the corresponding rate with the same interval in the reserved bounding

rates $(r_1, r_2, ..., r_P)_{reserved}$, a renegotiation request for more resources is initiated. The renegotiation request includes a set of new bounding rates $(r_1, r_2, ..., r_P)_{new}$ that are calculated by multiplying each bounding rate in the measured rates by α $(\alpha \geq 1)$. If the link control module determines there are enough network resources for the excess resource requirements of the renegotiation request, a successful confirmation is sent back to the stream control module, and the updated bounding rates $(r_1, r_2, ..., r_P)_{new}$ replace the reserved bounding rates of that connection. Otherwise, the connection has to proceed according to the previously reserved bounding rates $(r_1, r_2, ..., r_P)_{reserved}$. The procedure is illustrated in figure 2.2.2.1.

```
If ( any rate in (r_1, r_2, ..., r_P)_measured > any rate in (r_1, r_2, ..., r_P)_reserved )
{
        (r_1, r_2, ..., r_P)_new = α × max ( (r_1, r_2, ..., r_P)_measured , (r_1, r_2, ..., r_P)_reserved )
        Issue a renegotiation request for more resources;
        If ( admitted )
        {
                (r_1, r_2, ..., r_P)_reserved = (r_1, r_2, ..., r_P)_new ;
                Transmit at the updated (r_1, r_2, ..., r_P)_reserved ;
        }
        Else if ( rejected/blocked )
        {
                Transmit at the previous (r_1, r_2, ..., r_P)_reserved ;
        }
}
```

Figure 2.2.2.1: Segmentation procedure for requests for more resources

- Procedure for renegotiation requests for resource release

Since the bounding rates over multiple interval lengths are the maximum average rates as introduced in section 2.1, the last bounding rate r_P reflects the maximum allowed burst capacity over the longest interval length I_P. So the segmentation algorithm for the resource release request needs to only compare the r_P in the measured bounding rates $(r_1, r_2, ..., r_P)_{measured}$ with the r_P in the reserved bounding

rates $(r_1, r_2, ..., r_P)_{reserved}$. If the r_P in the measured rates $(r_1, r_2, ..., r_P)_{measured}$ falls below the r_P in the reserved rates $(r_1, r_2, ..., r_P)_{reserved}$ by a factor of β $(0 \le \beta \le 1)$, and at the same time there are at least MIN_RENE frames since last renegotiation point, the renegotiation for less resources immediately takes place, and the redundant network resources are released. The new bounding rates $(r_1, r_2, ..., r_P)_{new}$, that are the average of each bounding rate in the reserved rates $(r_1, r_2, ..., r_P)_{reserved}$ and those in the measured rates $(r_1, r_2, ..., r_P)_{measured}$, are reserved for the connection. MIN_RENE is applied here to avoid frequent resource release renegotiations. The procedure is illustrated in figure 2.2.2.2.

If (r_P in $(r_1, r_2, ..., r_P)_{measured}$ $< \beta \times$ r_P in $(r_1, r_2, ..., r_P)_{reserved}$)
AND the renegotiation interval since last renegotiation $\ge$ MIN_RENE)
{
$(r_1, r_2, ..., r_P)_{new} = 1/2\ (r_1, r_2, ..., r_P)_{measured} + 1/2\ (r_1, r_2, ..., r_P)_{reserved}$;
Issue a renegotiation request for resource release;
$(r_1, r_2, ..., r_P)_{reserved} = (r_1, r_2, ..., r_P)_{new}$;
Transmit at the updated $(r_1, r_2, ..., r_P)_{reserved}$;
}

Figure 2.2.2.2: Segmentation procedure for requests for resource release

Parameter α for the renegotiation for more resources is used to assign a little bit more network resources to the source than its actual requirement, which may help avoid consecutive renegotiations for more resources. Parameter β is applied to the renegotiation for resource release for an equivalent purpose. Therefore, the value of the (α, β) pair will affect the renegotiation frequency and network utilization.

2.2.3 Admission Control and Renegotiation Control Algorithms

Recall the discussion in section 2.2.1: the RED-VBR scheme provides a statistical service guarantee on top of a deterministic service. In response to these two services, two types of control mechanisms are provided in RED-VBR, connection admission control and renegotiation control. While the two control mechanisms seem to have similar functionalities, determining if there are sufficient resources to accommodate new resource requirements without degrading the QoS of the currently serviced connections, they actually work at different levels. The connection admission control works on the connection level to determine if a new connection can be admitted, while the renegotiation control works on the segment level to decide if a renegotiation request can be satisfied. For simplicity, the same algorithm is used both for the connection admission control and for the renegotiation control in our research.

Recent studies [5, 12, 13, 14, 15] show that admission control for deterministic service relies on delay analysis techniques, that is, the delay bound provided to a connection is a function of link speed and traffic constraint functions of all admitted connections. Denote d as the delay time, c as the link speed and $B_{R_T}^{j}(t)$ as the traffic constraint function for connection j, j = 1, 2, … , N. For a first come first serve (FCFS) scheduler, the upper bound on delay for all connections is given below:

$$d = \frac{1}{c} \max_{t \in t_k} \{ \sum_{j=1}^{N} B_{R_T}^{j}(t) - ct \}, \quad \text{j = 1, 2, …, N} \qquad (2.2.3\text{-}1)$$

Where the D-BIND traffic constraint function is

$$B_{R_T} = r_k \times t_k, \quad \text{k = 1, 2, … , P} \qquad (2.2.3\text{-}2)$$

Where r_k and t_k correspond to the D-BIND bounding rate and the interval length, respectively.

Whenever a renegotiation request is received at the link control module, the renegotiation control algorithm is immediately activated. It will first compute the current network

buffer requirement $Q(t)$ according to the previously reserved bounding rates $(r_1, r_2, ..., r_P)_{reserved}$ of other admitted sources as well as the currently required bounding rates $(r_1, r_2, ..., r_P)_{new}$ of source j. Based on the network buffer capacity Q ($Q = c \times d$) and currently occupied network buffer capacity $O(t)$, the currently available network capacity is obtained by $Q - O(t)$. The renegotiation control algorithm then compares the currently available network capacity $Q - O(t)$ with the current network buffer requirement $Q(t)$ to determine if the request of source j should be accepted or denied. The decision is finally sent back to the stream control module for source j. The detailed procedure is illustrated in figure 2.2.3.1, similar to [7, 8].

1. Initialize: $Q = c \times d$;
2. If (a new renegotiation request from source j arrives)

   {

   1. Calculate the current network buffer requirement $Q(t)$:

   $$Q(t) = \max\{0, \max_{1 \le k \le P}\{t_k (\sum_{i=1}^{N} a_i \times r_k in(r_1, r_2, ..., r_P)^i_{reserved} + r_k in(r_1, r_2, ..., r_P)^j_{new} - c)\}\}$$

   $a_i = 1$ if the ith source is accepted and $i \neq j$; Otherwise, $a_i = 0$;

   2. Calculate the currently available network buffer = $Q - O(t)$;
   3. If ($Q(t) \le Q - O(t)$)

      {

      Send an ‘admit’ response to source j;

      Update the reserved bounding rates of source j:

      $(r_1, r_2, ..., r_P)^j_{reserved} = (r_1, r_2, ..., r_P)^j_{new}$;

      }

      Else

      { Send a ‘reject’ response to source j; }

   }

Figure 2.2.3.1: Procedure of implementing renegotiation control algorithm

2.3 Issues observed from the experiments

The published research [5, 7, 8] verified/tested the proposed R-VBR schemes by experiments using one or two video traces. These video traces are encoded in either MPEG-1 or MPEG-2 coding schemes. We think that substantially more experiments on a variety of video traces would be helpful in rigorously investigating and evaluating the advantages and disadvantages of any R-VBR scheme. This is because different VBR compressed video traffic exhibits diverse burstiness properties over multiple time scales. For example, low quality VBR compressed video is much more bursty than high quality VBR compressed video. Therefore, we perform experiments on numerous video traces. All video traces used are MPEG-4 encoded real-world videos [16]. These video traces are either high quality encoded or low quality encoded.

Experiments with RED-VBR are conducted on more than ten video traces. For each experiment, the multiplexed video streams come from a single video trace. All the experiments employ the same simulation setup:

- The delay bound is set to be 0.2s;
- The D-BIND window size M is set to be 36 frames, where M is the number of consecutive frames to be used to compute the D-BIND bounding rates;
- $(\alpha = 1.1, \beta = 0.9)$;
- Each admitted connection is policed to transmit frames according to the reserved D-BIND bounding rates.

Based on the simulation results, the following phenomenon is observed. Some video traces can be perfectly policed according to the reserved rates. In other words, the dropped bits can be observed only in the policer. However, for some other video traces, the dropped bits may be observed both in the policer and in the network buffer sink (Note that a network buffer sink was set to collect the unexpected drops from the outgoing buffer). These unexpected drops probably occur because the segmentation algorithm

does not cooperate well with the renegotiation control algorithm. The following example helps explain this issue. In this example, there are no renegotiation requests for resource release.

Suppose:

Frame sequence: (50,000, 30,000, 40,000) bits;

Initial network occupancy = 0 bits;

Network buffer capacity = 40,000 bits;

Link processing capacity per frame time = 20,000 bits;

Two bounding bits are chosen (Note: for simplicity, the bounding rates are converted into the bounding bits by multiplexing the corresponding frame times).

Calculate the D-BIND bounding bits:

The reserved bounding bits over one frame time = 50,000 bits (frame size of the first frame);

The reserved bounding bits over consecutive two frame times = 80,000 bits (sum of frame sizes of the first two frames).

Conduct the renegotiation control test as described in section 2.2.3:

For one frame time: $50,000 - 20,000 \leq 40,000 - 0$;

For consecutive two frame times: $80,000 - 20,000 * 2 \leq 40,000 - 0$;

Therefore, the frame sequence can be admitted into the network buffer without loss.

Actual transmission procedure:

At the end of one frame time, the buffer occupancy = 50,000 + 0 – 20,000 = 30,000 bits, which is less than the network buffer capacity of 40,000 bits;

At the end of two frame times, the buffer occupancy = 30,000 + 30,000 – 20,000 = 40,000 bits, which is equal to the network buffer capacity of 40,000 bits;

At the end of three frame times, the buffer occupancy = 40,000 + 40,000 – 20,000 = 60,000 bits, which is more than the network buffer capacity of 40,000 bits.

Result:

Additional 20,000 bits have to be dropped.

According to the definition of the segmentation algorithm of RED-VBR in section 2.2.2, the renegotiation request for more resources is generated only when any rate in the measured bounding rates is more than the corresponding rate in the reserved bounding rates. Meanwhile, according to the renegotiation control algorithm discussed in section 2.2.3, the link control module conducts renegotiation control to allocate resources only when receiving a renegotiation request from any of the stream control modules. Otherwise, the link control module reserves the resources according to the reserved bounding rates of the admitted sources. In the above example, all the frames in the frame sequence are accurately bounded by the two specified D-BIND bounding bits, and thus the renegotiation request cannot be created. The link control module does not perform the renegotiation control during this period. However, additional 20,000 bits are dropped at the end of the three frame times. The occurrence of the unexpected drops is because the renegotiation control condition becomes unsatisfied during this period. This is because the network buffer occupancy is increased by the gradually filled bits as the frame time goes by. Therefore, although all the frames are transmitted according to the reserved bounding bits, there is a good possibility that the renegotiation control condition may become unsatisfied after several frame times due to the varying network buffer occupancy.

The above issue may be addressed to some extent by using the D-BIND traffic model and the (α, β) pair in the segmentation algorithm of RED-VBR. Compared with the actually required bandwidth by connections, RED-VBR reserves more bandwidth by α and releases the bandwidth only when the measured bandwidth is less than the reserved bandwidth by β. In addition to effectively avoiding consecutive renegotiations, the use of (α, β) pair can help decrease some of the unexpected drops. According to our observation, the number of unexpected drops is video trace dependent for a fixed D-BIND window size of 36 frames. In other words, the size of the D-BIND window M has an impact on the number of unexpected drops.

There are two ways to eliminate the unexpected drops. One approach would be that the link control module itself actively checks the renegotiation control condition frame by

frame based on the currently available network resources. Under this condition, whenever the renegotiation control condition is broken, the link control module can immediately inform each connection and decrease their reserved bounding rates. However, this approach may lead to fairly heavy control overhead. The other approach is to increase the D-BIND window size M until the unexpected drops are eliminated. Such a smallest sufficient D-BIND window size is defined to be the minimal D-BIND window size $M_{\min}$. A D-BIND window size M, smaller than $M_{\min}$, cannot completely capture the traffic characteristics of a video trace, and therefore would cause some unexpected drops in the network buffer sink. On the other hand, if M is much greater than $M_{\min}$, it will cause unnecessary computational complexity because of the increased D-BIND window size. Table 2.3.1 presents the value of $M_{\min}$ for various video traces. The encoded video traces include several types, such as movies, cartoons, sports events and TV sequences. Experiments are tested under the following conditions:

- The delay bound is set to be 0.2 s;
- $(\alpha = 1.1, \beta = 0.9)$;
- The simulation time is 40,000 frames (approximately 26 minutes);
- Each admitted connection is policed to transmit frames according to the reserved bounding rates;
- The bounding rates are continuously computed from one frame time to P frame times, with a step size of one frame time, where $P = M_{\min}$.

QoS	Video Trace	D-BIND minimum window size (M_{min}) (Frame)
High Quality	Silence of the Lambs	36
	Star Wars IV	36
	Mr. Bean	36
	First Contact	60
	Soccer	48
	ARD News	72
	ARD Talk	>72
	Lecture Room	24
Low Quality	Silence of the Lambs	36
	Star Wars IV	24
	Mr. Bean	36
	Robin Hood	48
	First Contact	36

Table 2.3.1: Overview of the minimal D-BIND window size

As summarized in table 2.3.1, most video traces have the value of M_{min} around 36 frames. The value of M_{min} for *ARD News* and *ARD Talk*, however, is larger than 60 frames. This indicates that these two video traces are more bursty and the bursts can last for a long time.

The admission control condition (Equation 2.2.3-1 in section 2.2.3) indicates that the number of admissible connections is a function of the delay bound and of the traffic constraint functions for all the connections. For a given delay bound, if each connection can be tightly bounded, more connections may be admitted. Using the tightest D-BIND traffic constraint function for each connection would lead to better utilization of network resources. As discussed in section 2.1, the tightest D-BIND traffic constraint function is constructed by the consecutive P rate-interval pairs where P is equal to the D-BIND window size M. In order to eliminate the unexpected drops, M would be equal to or more

than $M_{\min}$. The large size of M means that more bounding rates are needed. This would greatly increase the processing complexity and memory space in the following aspects:

- Computing the D-BIND bounding rates. The D-BIND traffic model calculates the bounding rates from the interval length of one frame time to P frame times continuously, and each bounding rate is computed by looping through the segment with M frames. The large size of P directly leads to more loops. In addition, the fact that the segmentation algorithm of RED-VBR updates the bounding rates frame by frame further increases the computation volume.
- Carrying more parameters by renegotiation requests.
- Performing renegotiation control. The renegotiation control algorithm has to determine if there are enough resources from interval length of one frame time to P frame times.

2.4 A new approach: Virtual-Queue-Based RED-VBR

In order to simplify the computational complexity of RED-VBR while obtaining relatively high network utilization, we propose a new solution, named Virtual-Queue-Based RED-VBR. This solution introduces a concept of virtual network buffer into RED-VBR. The virtual network buffer is only for the use of the admission control and renegotiation control conditions.

The size of the virtual network buffer ($Q_{virtual}$) will be less than or equal to the actual size of the network buffer ($Q = c \times d$). A corresponding parameter λ $(0 \le \lambda \le 1)$ is used here to represent the ratio of virtual network buffer capacity ($Q_{virtual}$) to the actual network buffer capacity (Q), and is named a tuning parameter. The value of λ is set to be 1 by default. When unexpected drops are observed in the network buffer sink, the value of λ is

decreased to adjust the virtual network buffer capacity ($Q_{virtual}$). This method actually decreases the available network buffer capacity ($Q - O(t)$) for the check of admission and renegotiation control conditions. Such a limitation on the available network buffer capacity can help eliminate the unexpected drops and reduce the complexity of D-BIND model used in RED-VBR. Note that RED-VBR empirically finds out $M_{\min}$ first and then sets $M \geq M_{\min}$ to ensure appropriate use of the D-BIND traffic model. Unlike that scheme, Virtual-Queue-Based RED-VBR imposes no limit on the D-BIND window size M. In addition, this approach can use a smaller size of P (smaller than M) with the introduced virtual queue (i.e., through adjusting the tuning parameter λ) to achieve network performance comparable to RED-VBR (P is equal to M). This scheme will greatly decrease the computational complexity and memory space needed in RED-VBR.

For a given delay bound, decreasing the value of λ can gradually reduce the unexpected drops in the network buffer sink. Until the unexpected drops are totally eliminated in the network buffer sink, the λ at this point is defined to be the maximum tuning parameter $\lambda_{\max}$. Two groups of experiments are conducted to test $\lambda_{\max}$ for the delay bound varying from 0s to 0.2s. Both experiments are tested under the following conditions: 1) P is set to be 12, the size of one GOP; 2) the value of M is video trace dependent and $M = M_{\min}$; 3) $(\alpha = 1.3, \beta = 0.7)$.

In the first group of experiments, each experiment is conducted on homogeneous multiplexed video streams with independently starting at a random place of the tested video trace. Figure 2.4.1 depicts the value of $\lambda_{\max}$ as a function of delay bound.

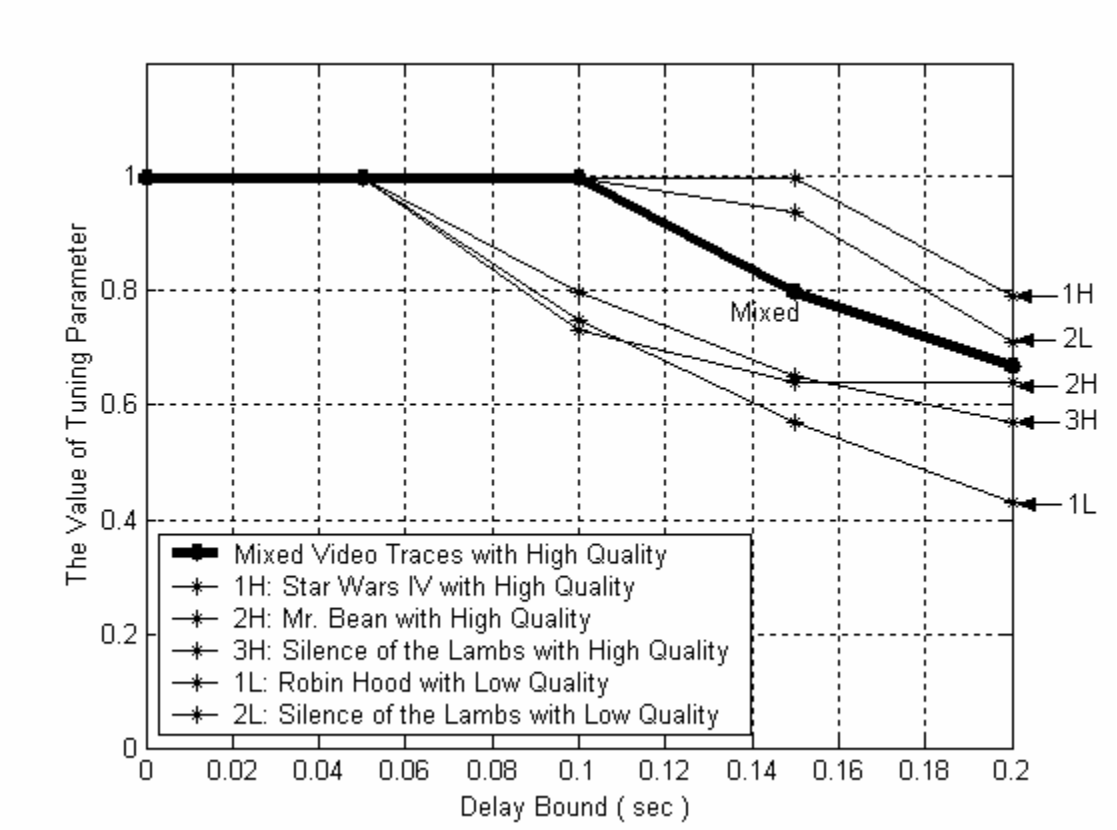
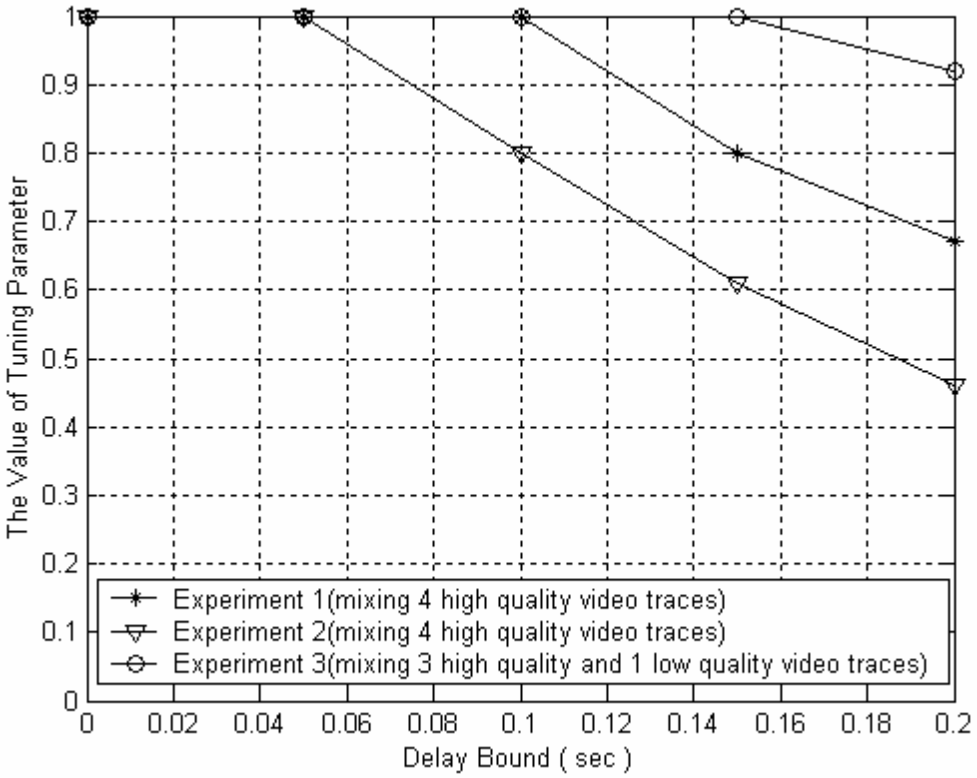


Figure 2.4.1: Tuning parameter λ_{max} of a single video trace

As shown in figure 2.4.1, λ_{max} of various video traces exhibits a similar decaying pattern. When the delay bound is less than 0.05s, λ_{max} equals 1 for all video traces. This indicates that the segmentation algorithm and renegotiation control algorithm can cooperate very well and the frames transmitted according to the reserved bandwidth can be successfully fed into the network buffer. As the delay bound increases from 0.05s to 0.2s, the value of λ_{max} begins to decrease (from 1) to reduce the actually available resources to eliminate the unexpected drops. The delay bound corresponding to λ_{max} is defined as the maximum delay bound d_{max} . The bold curve represents the result for the experiment in which heterogeneous multiplexed video streams are randomly chosen from the four high quality video (i.e., experiment 1 in the second group of the experiments). The value of d_{max} of this experiment is 0.1s, longer than d_{max} = 0.05s for *Mr. Bean* and *Silence of the Lambs* and shorter than d_{max} = 0.15s for *Star Wars IV*.

In the second group of experiments, each experiment selects four different video traces to generate heterogeneous multiplexed video streams. All the video traces in experiment 1 and experiment 2 are from the high quality encoding. The video traces in experiment 3 include one high quality video trace and three low quality video traces.



	Video Trace Name
Experiment 1	High quality: Star Wars IV, Mr. Bean, Silence of the Lambs, Soccer
Experiment 2	High quality: Robin Hood, First Contact, Susi und Strolch(Cartoon), ARD talk
Experiment 3	High quality: Star Wars IV, Mr. Bean, Susi und Strolch(Cartoon) Low quality: Robin Hood

Figure 2.4.2: Tuning parameter λ_{max} of mixed video traces

As indicated in figure 2.4.2, the values of λ_{max} of the three experiments exhibit a similar decaying tendency again, that is, λ_{max} is always equal to 1 when the delay bound is

shorter than $d_{\max}$ and approximately linearly decreases for the delay bound longer than $d_{\max}$. This tendency may be expressed by an empirical equation below:

$$\lambda_{\max} = \begin{cases} 1 & 0 \le d \le d_{\max} \\ -4d + 4d_{\max} + 1 & d_{\max} \le d \le 0.2 \end{cases} \qquad (2.4\text{-}1)$$

where $d_{\max}$ is the maximum delay bound when the maximum tuning parameter $\lambda_{\max}$ begins to decrease from the value of 1. $d_{\max}$ is dependent on the characteristics of the multiplexed video streams, and may be estimated by some online method, which will be studied in our future work.

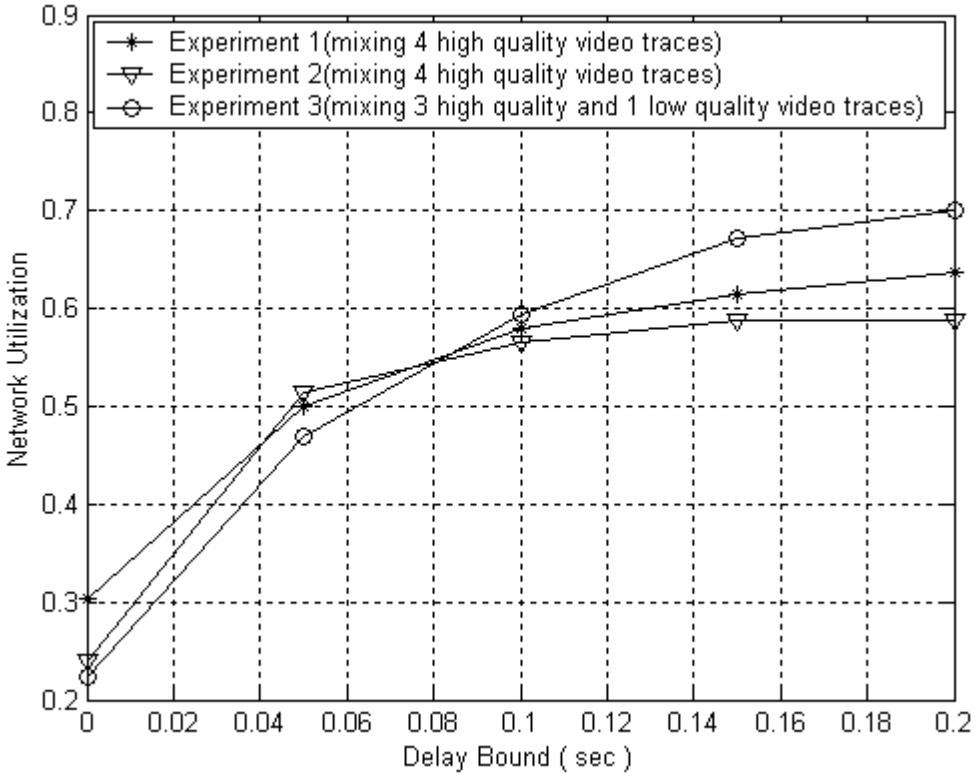


Figure 2.4.3: Network utilization of mixed video traces

Furthermore, we observe from figure 2.4.1 that the unexpected drops for experiment 3 cannot be observed until the delay bound is beyond 0.15s. Therefore, as the delay bound increases, multiplexing high quality and low quality video traces may help explore the potential SMG among the admitted video streams better. Compared to the other two

experiments of mixed high quality video traces, higher network utilization can be expected for experiment 3 as illustrated in figure 2.4.3.

2.5 Experiment Setup

Our research is a simulation-oriented study. Trace-driven simulations are used to evaluate network performance on different scenarios. We conduct our experiments on MPEG-4 encoded video traces due to their wide range of bit rates, from very low bit rates used in wireless communications to very high bit rates applied in high definition television (HDTV). All the MPEG-4 encoded frame-size video traces are chosen from a publicly available library [16], generated at the Technical University Berlin. The MPEG-4 encoded video traces are generally 60 minutes long (around 89,999 frames) with a GOP pattern of IBBPBBPBBPBB and a frame rate of 25 frames / second. The total simulation time is 40,000 frames (about 0.5 hours). Each source starts at an independently random frame within the first 20,000 frames of an entire trace.

The basic experiment setups are listed below:

- Video traces are chosen from various sources, such as movies, sports events and cartoons. These traces are either high quality encoded or low quality encoded.
- The video streams are multiplexed over a single communication line using a FCFS scheduler.
- The multiplexed video streams for a single experiment are grouped in two ways.
    - The multiplexed video streams are from a single video trace.
        - If this is a high quality video trace, the link speed c = 45 Mbps.
        - If this is a low quality video trace, the link speed c = 11 Mbps.
    - The multiplexed video streams are from multiple video traces (i.e., heterogeneous video traces). The link speed c = 45 Mbps.
- For simplicity, renegotiation control uses the same algorithm as admission control, as discussed in section 2.2.3.

- To deal with the rejected renegotiation requests, some researchers [7, 8] choose to block the sources whose request is rejected until their next renegotiation points. Our research chooses to perform traffic shaping instead. The rejected video streams will be transmitted according to the reserved bounding rates. Note that this denial will not affect the next renegotiation request, which is generated only according to the segmentation algorithm.
- The delay bound ranges from 0s to 0.2s with a step size of 0.05s. For each delay bound, the corresponding performance metrics are collected.
- The collected performance metrics are:
    - Average renegotiation blocking probability refers to the probability that a resource request is rejected by the link control module. It is calculated as the ratio of the accumulated time for which renegotiation requests for more bandwidth have been denied to the total connection time.
    - Average network utilization is decided by dividing the connection's long-term-average rate by the link speed. The long-term-average rate is defined as the total number of bits transmitted over the connection lifetime.
    - Average renegotiation interval is determined by dividing the connection lifetime by the number of renegotiations.
    - Average drop rate is defined to be the portion of dropped bits, due to applying traffic shaping, over the total number of bits that the video stream needs to transmit.
- The D-BIND window size M is video trace dependent and generally set to be multiple times of one GoP (12 frames).
    - With Virtual-Queue-Based RED-VBR, P = 12.
    - With RED-VBR, P = $M_{\min}$.

The implementation procedure similar to the description in [7] is shown in figure 2.6.1.

1. Initialize: set the renegotiation blocking probability to 0.01;
   set MIN_RENE = 30 frames;
   set the delay bound and the initial number of multiplexed sources;
   set other parameters according to different scenarios;
2. Connection set up sub-process ( )
   Call process of computing traffic parameters;
   Call process of admission control algorithm;
   If (the connection is admitted) { Call renegotiation sub-process; }
   Else { Try again; }
3. Renegotiation sub-process ( )
   Call process of computing traffic parameters;
   Call process of online segmentation algorithm;
   If (a new renegotiation request is generated)
   {
   Call process of renegotiation control algorithm;
   If (the request is accepted)
   { Transmit according to the updated $(r_1, r_2, ..., r_P)_{reserved}$; }
   Else { Transmit according to the previous $(r_1, r_2, ..., r_P)_{reserved}$; }
   }
   Repeat the renegotiation sub-process until the end of simulation;
4. If (blocking probability ≤ 0.01)
   { Increase number of sources;
   Run simulations from 2; }
   Else
   { Decrease number of sources;
   Run simulations from 2; }
5. Increase the delay bound and start from 1;
6. End.

Figure 2.5.1: Main procedure of trace-driven simulation

2.6 Experiment Results and Analysis

We focus on the performance evaluation of the proposed Virtual-Queue-Based RED-VBR compared to RED-VBR and Overlap-Based RED-VBR introduced in [11]. The impacts of each performance metric are discussed in detail based on the simulation results.

A Brief Review of Overlap Based RED-VBR

An approximation D-BIND traffic model has also been proposed along with the D-BIND traffic model [11]. We refer to RED-VBR in which the D-BIND traffic model is replaced with the approximation D-BIND traffic model as Overlap-Based RED-VBR. This scheme approximately describes a video sequence with a D-BIND window size M by P D-BIND rate-interval pairs, where $P < M$. The difference between Virtual-Queue-Based RED-VBR and Overlap-Based RED-VBR is in executing the admission control and renegotiation control algorithms. Overlap-Based RED-VBR still uses M bounding rates to build the traffic constraint function for the admission and renegotiation control conditions. Those bounding rates for an interval length longer than P are derived by an overlapping approximation approach. This approach first builds the traffic constraint function $B_{R_T}[t_1,...,t_P]$ from P rate-interval pairs carried in a renegotiation request. Then, the traffic constraint function $B_{R_T}[t_{P+1},...,t_M]$ for an interval length longer than P is approximately derived by overlapping $B_{R_T}[t_1,...,t_P]$ several times, as demonstrated in figure 2.6.1.1. Note that the GOP pattern in the plot is IBBPBB. The bounding rates $(r_{P+1}, r_{P+2},..., r_M)$ for an interval length longer than P are finally computed from $B_{R_T}[t_{P+1},...,t_M]$. The detailed algorithm is depicted in figure 2.6.1.2.

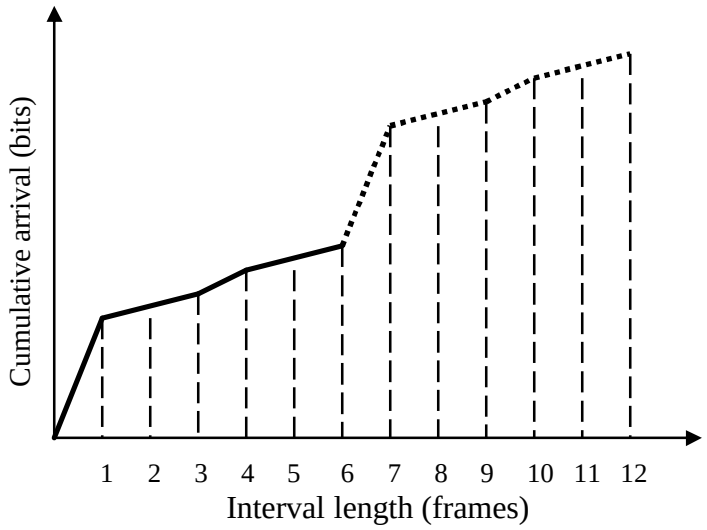


Figure 2.6.1.1: Overlapping approximation approach to build traffic constraint function

1. Use $(r_1, r_2, ..., r_P)$ to construct the traffic constraint function $B_{R_T}[t_1, ..., t_P]$.
2. Derive the traffic constraint function $B_{R_T}[t_{P+1}, ..., t_M]$ by overlapping $B_{R_T}[t_1, ..., t_P]$ several times.
3. Calculate $(r_{P+1}, r_{P+2}, ..., r_M)$ from $B_{R_T}[t_{P+1}, ..., t_M]$
4. Reconstruct $(r_1, r_2, ..., r_M)$

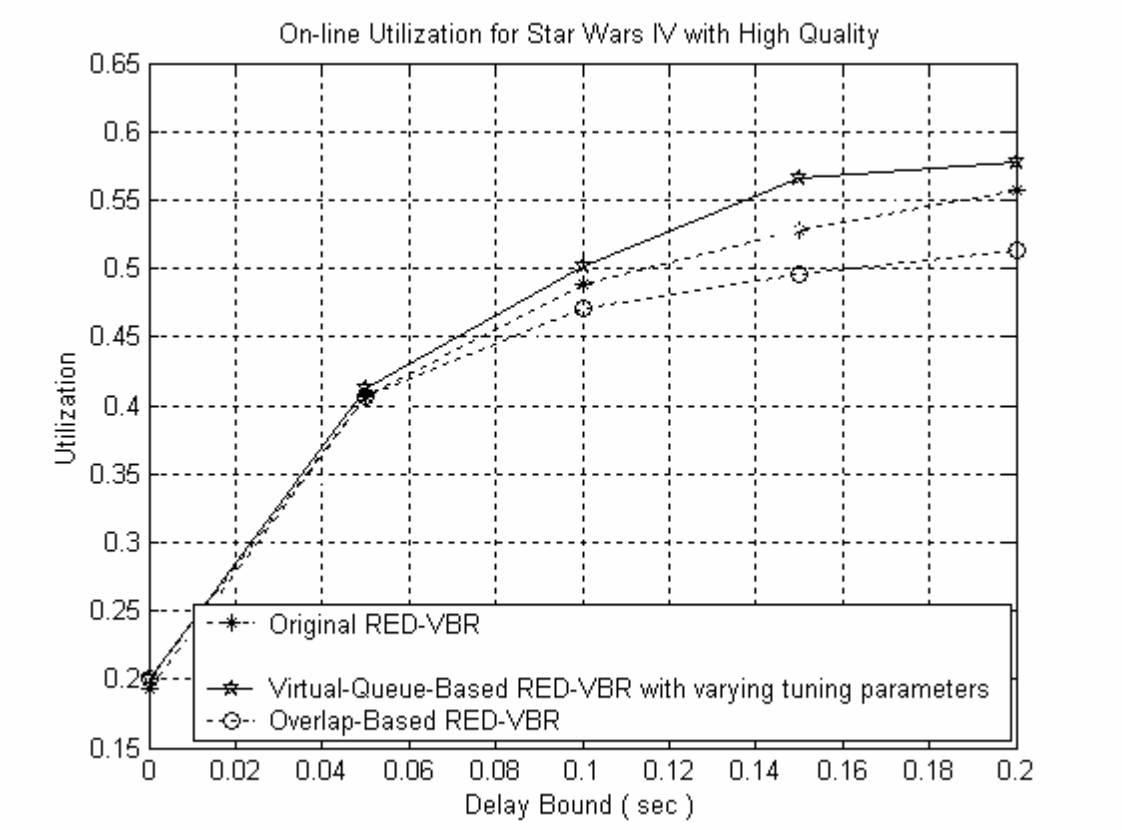
Figure 2.6.1.2: Process of overlapping approximation approach

Since Overlap-Based RED-VBR uses M rate-interval pairs to perform the admission control and renegotiation control algorithms, this scheme does not reduce the computational complexity involving in the link control module. Meanwhile, similar to RED-VBR, Overlap-Based-RED-VBR has to set $M \geq M_{\min}$ to eliminate the unexpected drops as well. On the other hand, since the traffic constraint function ($B_{R_T}[t_{P+1}, ..., t_M]$) for an interval length longer than P is derived through the overlapping approximation, this function is not the tightest D-BIND traffic constraint function. Overlap-Based RED-VBR may therefore achieve relatively lower network utilization than RED-VBR.

2.6.1 Performance Comparison among Virtual-Queue-Based RED-VBR, RED-VBR and Overlap-Based RED-VBR

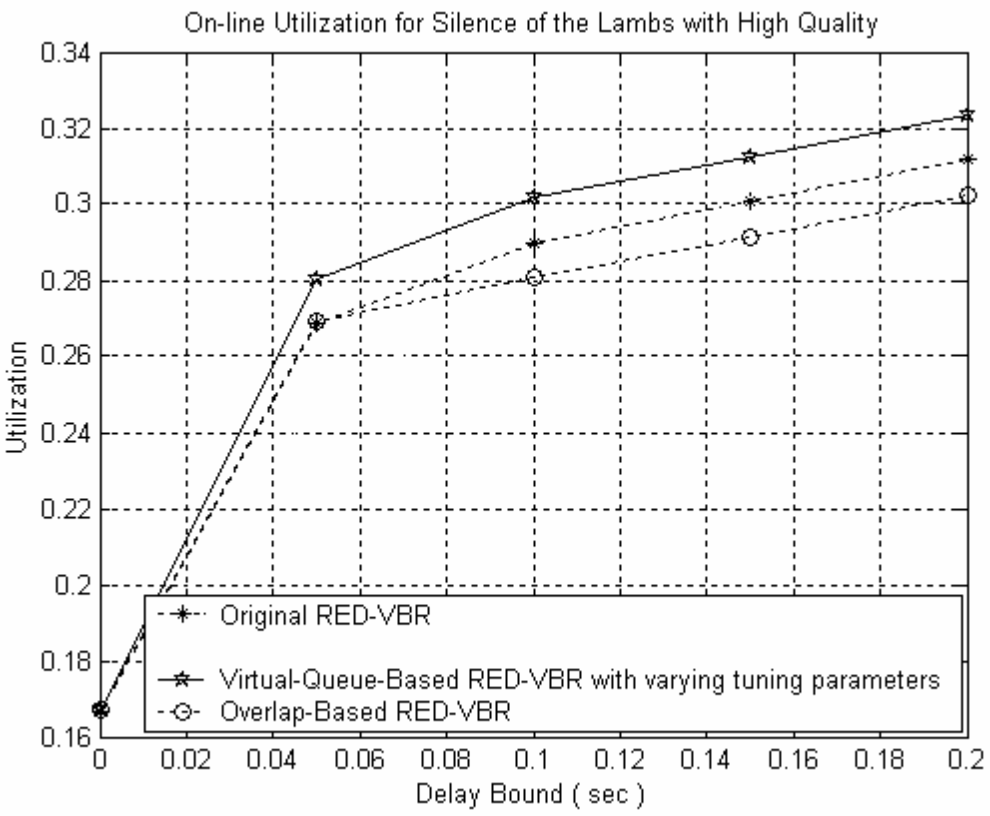
For RED-VBR and Overlap-Based RED-VBR, we empirically determine $M_{\min}$ for a video trace with delay bound of 0.2s and then use this $M_{\min}$ to conduct experiments for delay bounds varying from 0s to 0.15s. For ease of comparison, Virtual-Queue-Based RED-VBR uses the same $M_{\min}$ for this video trace as the other two schemes do. For Virtual-Queue-Based RED-VBR, the performance metrics are collected when $\lambda = \lambda_{\max}$ for each delay bound. The performance comparison is conducted based on six experiments. The multiplexed video streams for five experiments are from a single video trace. They are three high quality video traces (*Star Wars IV, Silence of the Lambs* and *Mr. Bean*) and two low quality video traces (*Robin Hood* and *Silence of the Lambs*). The sixth experiment is tested on mixing four types of high quality vide traces (*Star Wars IV*, *Silence of the Lambs*, *Mr. Bean* and *Soccer*). The simulation results for network utilization are depicted in figure 2.6.1.3 to figure 2.6.1.8. A table is added for each plot to show the traffic statistics (data come from [16]) and some key performance parameters from the simulation results. In these tables,

- The smoothing ratio is: $\left(\frac{Peak}{Mean}(Frame) - \frac{Peak}{Mean}(GoP)\right) \Big/ \frac{Peak}{Mean}(Frame)$. (2.6.1-1)
- The network utilization is collected with delay bound of 0.2s.
- The drop rate and renegotiation interval are averaged for delay bounds from 0s to 0.2s.



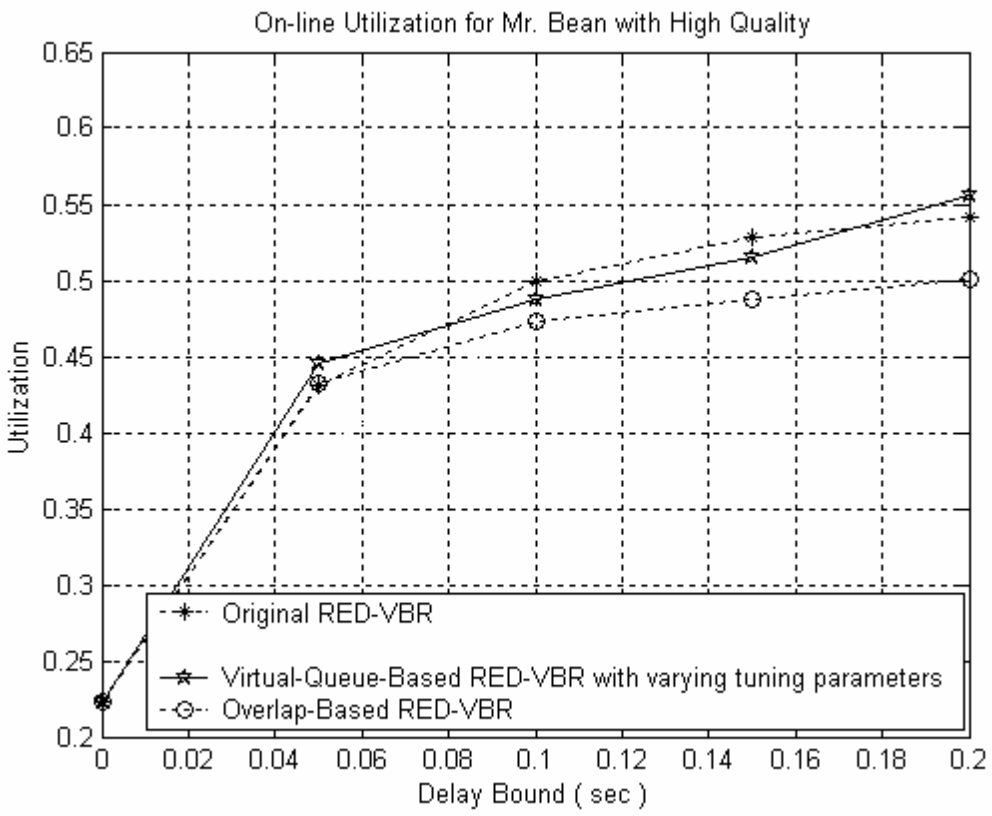
Video Trace	Star Wars IV			
Traffic Statistics	Quality	Frame Statistics Peak/Mean	GOP Statistics Peak/Mean	Smoothing Ratio
	High	6.81	4.29	37%
Performance Metrics	Scheme	Network Utilization (delay = 0.2s)	Average Drop rate	Average Rene-Interval (sec)
	RED-VBR	55.8%	0.70%	1.67
	Overlap-Based RED-VBR	51.3%	0.41%	1.98
	Virtual-Queue-Based RED-VBR	57.7%	0.42%	2.00

Figure 2.6.1.3: Online network performance of *Star Wars IV* with high quality



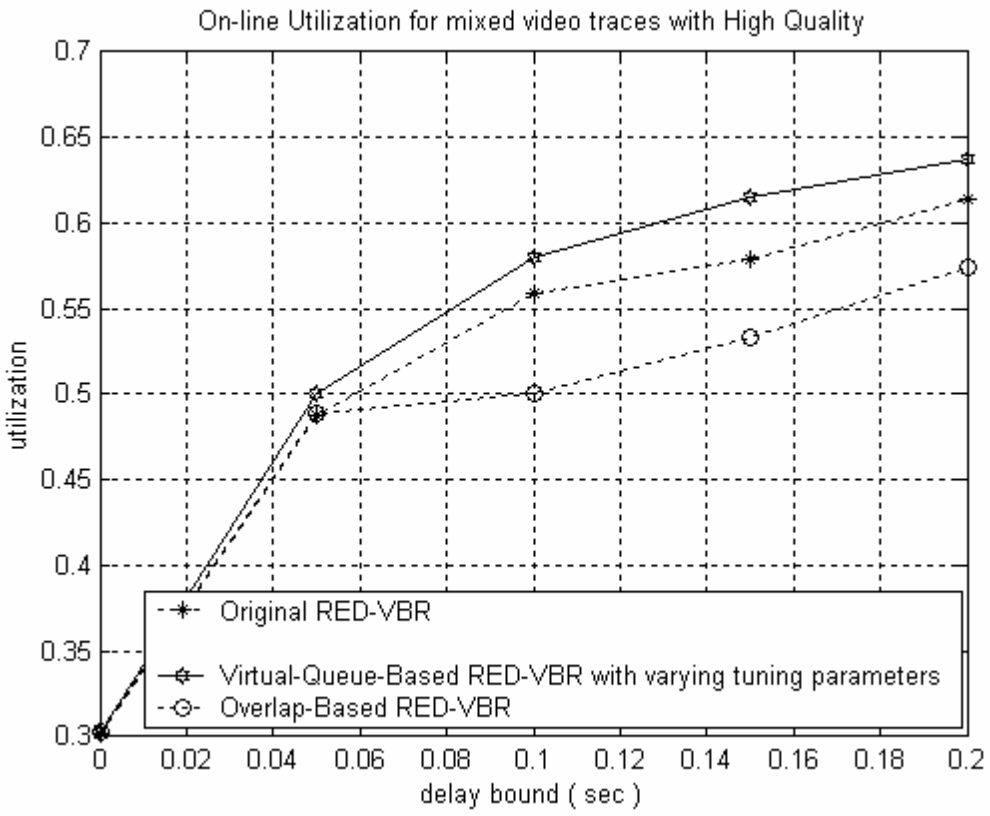
Video Trace	Silence of the Lambs			
Traffic Statistics	Quality	Frame Statistics Peak/Mean	GOP Statistics Peak/Mean	Smoothing Ratio
	High	7.73	6.22	19.5%
Performance Metrics	Scheme	Network Utilization (delay = 0.2s)	Average Drop rate	Average Rene-Interval (sec)
	RED-VBR	31.2%	1.07%	1.59
	Overlap-Based RED-VBR	30.2%	0.70%	1.90
	Virtual-Queue-Based RED-VBR	32.3%	0.81%	1.85

Figure 2.6.1.4: Online network performance of *Silence of the Lambs* with high quality



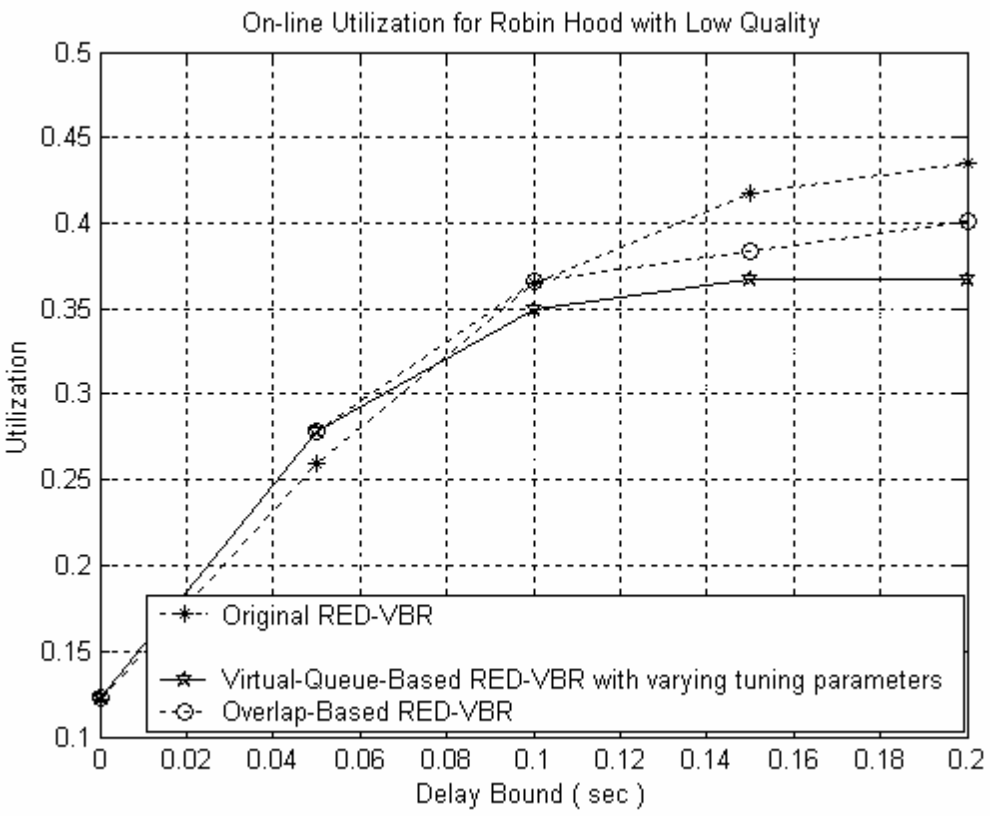
Video Trace	Mr. Bean			
Traffic Statistics	Quality	Frame Statistics Peak/Mean	GOP Statistics Peak/Mean	Soothing Ratio
	High	5.24	3.73	28.8%
Performance Metrics	Scheme	Network Utilization (delay = 0.2s)	Average Drop rate	Average Rene-Interval (sec)
	RED-VBR	54.2 %	0.58%	1.74
	Overlap-Based RED-VBR	50.1%	0.36%	2.02
	Virtual-Queue-Based RED-VBR	55.6%	0.40%	2.04

Figure 2.6.1.5: Online network performance of *Mr.Bean* with high quality



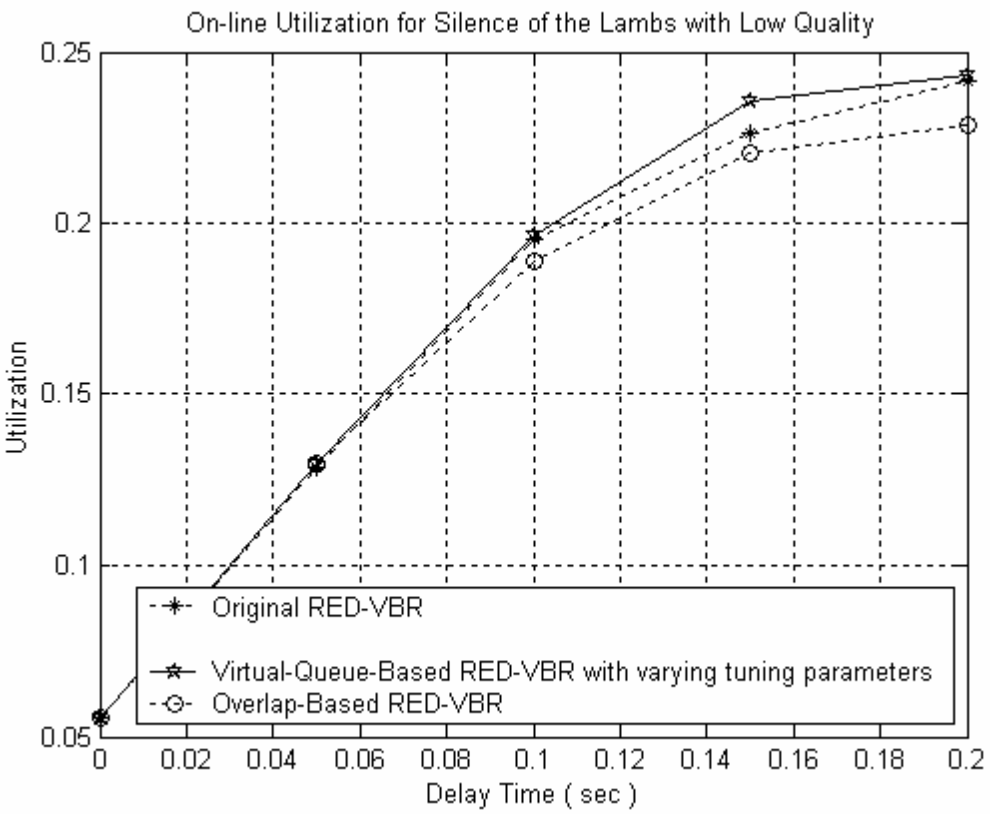
Video Trace	Star Wars IV, Mr. Bean, Silence of the Lambs, Soccer			
Traffic Statistics	Quality	Frame Statistics Peak/Mean	GOP Statistics Peak/Mean	Soothing Ratio
	High	\	\	\
Performance Metrics	Scheme	Network Utilization (delay = 0.2s)	Average Drop rate	Average Rene-Interval (sec)
	RED-VBR	61.4 %	0.72%	1.58
	Overlap-Based RED-VBR	57.4%	0.48%	1.82
	Virtual-Queue-Based RED-VBR	63.7%	0.53%	1.82

Figure 2.6.1.6: Online network performance of mixed video traces with high quality



Video Trace	Robin Hood			
Traffic Statistics	Quality	Frame Statistics Peak/Mean	GOP Statistics Peak/Mean	Smoothing Ratio
	Low	10.56	3.34	68.4%
Performance Metrics	Scheme	Network Utilization (delay = 0.2s)	Average Drop rate	Average Rene-Interval (sec)
	RED-VBR	43.5 %	0.93%	1.47
	Overlap-Based RED-VBR	40.1%	0.46%	1.89
	Virtual-Queue-Based RED-VBR	36.7%	0.44%	1.83

Figure 2.6.1.7: Online network performance of *Robin Hood* with low quality



Video Trace	Silence of the Lambs			
Traffic Statistics	Quality	Frame Statistics Peak/Mean	GOP Statistics Peak/Mean	Smoothing Ratio
	Low	21.39	10.48	51%
Performance Metrics	Scheme	Network Utilization (delay = 0.2s)	Average Drop rate	Average Rene-Interval (sec)
	RED-VBR	24.2 %	2.46%	1.33
	Overlap-Based RED-VBR	22.9%	1.69%	1.51
	Virtual-Queue-Based RED-VBR	24.3%	1.82%	1.48

Figure 2.6.1.8: Online network performance of *Silence of the Lambs* with low quality

Performance analysis by the statistics of video traces

Quality	Video Trace	Frame Statistics Peak/Mean A	GOP Statistics Peak/Mean B	Smoothing Ratio $\frac{A-B}{A}$	Network Utilization (delay bound = 0.2s)
High	Star Wars IV	6.81	4.29	37.0%	55.8%
	Silence of the Lambs	7.73	6.22	19.5%	31.2%
	Mr. Bean	5.24	3.73	28.8%	54.2%
Low	Robin Hood	10.56	3.34	68.4%	43.5%
	Silence of the Lambs	21.39	10.48	51.0%	24.2%

Table 2.6.1.1: Effect of smoothing ratio

From table 2.6.1.1, we see that the difference of achievable network utilization of various video traces is probably a consequence of the differing smoothing ratio. A smoothing ratio with a large value indicates that the bursts may be effectively smoothed out within less GOPs, and therefore relatively high network utilization could be achieved. For example, the smoothing ratio for the high quality video traces is 37% for *Star Wars IV*, 19.5% for *Silence of the Lambs*, and 28.8% for *Mr. Bean*. Therefore, *Star Wars IV* and *Mr. Bean* achieve a higher network utilization (around 55%) than *Silence of the Lambs*, which has network utilization of 31.2%. A similar tendency can be observed for the video traces with low quality. *Robin Hood* with a smoothing ratio of 68.4% achieves network utilization of 43.5%, which is higher than the network utilization of 24.2% of *Silence of the Lambs*, whose smoothing ratio is 51%. Virtual-Queue-Based RED-VBR and Overlap-Based RED-VBR exhibit a similar tendency as RED-VBR. Note that the relationship between the smoothing ratio and the achievable utilization is applicable within the same category, such as either for low quality video traces or for high quality video traces. In addition, for any renegotiation scheme, video traces with high quality seem to achieve much higher network utilization than video traces with low quality. This is possibly

because low quality video traces have a higher compression ratio than high quality video traces. Consequently, low quality video traces are more bursty than high quality video traces. This makes it difficult for video streams from a low quality video trace to extract high SMG among the bursty video streams.

Performance Analysis of Network Utilization

- Network utilization for high quality video traces

Overall, the three schemes show a similar pattern, that is, the network utilization increases as the delay bound increases. The network utilization shows a rapid rise for the delay bound from 0s to 0.05s, and a slow rise for the delay bound varying from 0.05s to 0.2s. This implies that more and more video streams can be simultaneously admitted into the network, but at the expense of a degraded QoS (i.e., a long delay).

Even though the network utilizations of the three schemes represent a similar upward trend, they increase at different rates depending on the delay bound. For a short delay bound (less than 0.05s), the three schemes achieve almost the same network utilization. This implies that for a small number of admitted video streams (due to the short delay bound), the segmentation algorithm can cooperate very well with the renegotiation control algorithm and the network buffer can efficiently absorb all the admitted video streams. As the delay bound increases, Virtual-Queue-Based RED-VBR generally tends to achieve higher network utilization than the other two schemes. The utilization improvement becomes prominent for delay bound greater than 0.15s for all video traces. By comparison, the network utilization of Overlap-Based RED-VBR increases more slowly with the delay bound. Therefore, the utilization difference between Virtual-Queue-Based RED-VBR and Overlap-Based gradually increases as the delay bound increases. For instance, for the delay bound of 0.2s, compared with RED-VBR, Virtual-Queue-Based RED-VBR has a network utilization improvement of approximately 3.4% for *Star Wars IV*, 3.5% for *Silence of the Lambs*, 2.6% for *Mr. Bean*. As compared with

Overlap-Based RED-VBR, the network utilization improvement with Virtual-Queue-Based RED-VBR is around 12.5% for *Star Wars IV*, 7% for *Silence of the Lambs*, and 11% for *Mr. Bean*. The relatively low utilization of Overlap-Based RED-VBR is probably due to the use of the approximation approach to calculate the D-BIND bounding rates as discussed before. Since the D-BIND traffic constraint function derived by the approximation approach is not the tightest one, the renegotiation control algorithm would therefore over allocate resources to each connection, thus leading to low utilization.

It is interesting to note that mixing different types of video traces is beneficial in improving the network utilization for any scheme, illustrated in figure 2.1.6.6. For any delay bound longer than 0s, the experiment using the heterogeneous video traces achieves higher network utilization than that using the homogeneous video traces. For instance, when the delay bound is 0.2s, the network utilization of the mixed video traces for Virtual-Queue-Based RED-VBR is around 63.7%, which is higher than 57.7% for *Star Wars IV*, 55.6% for *Mr. Bean*, and 32.3% for *Silence of the Lambs*. Moreover, we observe that the difference of the network utilization among the three schemes becomes more prominent. In other words, Virtual-Queue-Based RED-VBR achieves apparently more significant network utilization than the other two schemes for delay bounds from 0s to 0.2s.

- Network utilization for low quality video traces

As shown in figure 2.6.1.7 and 2.6.1.8, for low quality video traces, the three schemes achieve similar network utilization for delay bound less than 0.1s. As the delay bound increases, for *Silence of the Lambs*, Virtual-Queue-Based RED-VBR achieves higher network utilization than the other two schemes. However for *Robin Hood*, RED-VBR and Overlap-Based RED-VBR obtain better network utilization than Virtual-Queue-Based RED-VBR. Therefore, Virtual-Queue-Based RED-VBR does not work well for some low quality video traces. This issue needs to be further studied.

Performance Metrics: Average Renegotiation Interval and Average Drop Rate

In general, the estimation of static NRAM systems relies on link utilization. Unlike static NRAM systems, dynamic NRAM systems need to consider several other performance metrics besides link utilization, such as renegotiation interval and renegotiation blocking probability. As discussed in section 2.2, decreasing the renegotiation interval will increase the network utilization, at the expense of increased control overhead in the network. Any scheme in a dynamic NRAM system needs to achieve an efficient balance among these performance metrics.

Quality	Video Trace	Scheme	Network Utilization	Ave Drop rate	Ave Rene-Interval (s)
High	Star Wars IV	RED-VBR	55.8%	0.70%	1.67
		Overlap-Based	51.3%	0.41%	1.98
		Virtual-Queue-Based RED-VBR	57.7%	0.42%	2.00
	Mixed Video Traces	RED-VBR	61.4 %	0.72%	1.58
		Overlap-Based	57.4%	0.48%	1.82
		Virtual-Queue-Based RED-VBR	63.7%	0.53%	1.82
Low	Robin Hood	RED-VBR	43.5 %	0.93%	1.47
		Overlap-Based	40.1%	0.46%	1.89
		Virtual-Queue-Based RED-VBR	36.7%	0.44%	1.83

Table 2.6.1.2: Overview of performance metrics

Table 2.6.1.2 lists the average renegotiation interval and average drop rate collected from figure 2.6.1.3 to figure 2.6.1.8. We observe that Virtual-Queue-Based RED-VBR

achieves comparable average renegotiation interval and average drop rate as Overlap-Based RED-VBR. Compared with RED-VBR, Virtual-Queue-Based RED-VBR has longer average renegotiation interval and smaller average drop rate. For example, for low quality *Robin Hood*, the average drop rate reduces from 0.93% for RED-VBR to 0.44% for Virtual-Queue-Based RED-VBR, and the average renegotiation interval increases from 1.47s for RED-VBR to 1.83s for Virtual-Queue-Based RED-VBR. That is, the improvement by Virtual-Queue-Based RED-VBR on the average renegotiation interval ranges from around 11% to around 24%, and on the average drop rate ranges from around 24% to around 52%, with respect to RED-VBR. Therefore, Virtual-Queue-Based RED-VBR can achieve a better or comparable overall network performance with significantly lower computational complexity and simpler implementation as compared with RED-VBR and Overlap-Based RED-VBR. In the case of the experiments of high quality video traces, the overall performance improvement of Virtual-Queue-Based RED-VBR becomes prominent when compared with RED-VBR and Overlap-Based RED-VBR. That is, Virtual-Queue-Based RED-VBR can achieve higher network utilization, in addition to larger average renegotiation interval and smaller average drop rate than those of RED-VBR.

2.6.2 Other Investigations on RED-VBR

As discussed in section 2.2.2, the segmentation algorithm of RED-VBR can vary the value of (α, β) pair to correspondingly adjust the renegotiation interval. As α decreases from 1.3 to 1 and β increases from 0.7 to 1, the request for more resources would request the bandwidth which is close to the actual measured bandwidth and the request for resource release would more easily be generated whenever the measured bandwidth is close to the reserved bandwidth. Such a simultaneous adjustment on the value of (α, β) pair would accordingly decrease the renegotiation interval, allowing each admitted video stream to frequently extract the potential SMG. As a result, higher network utilization can be expected. The scenario with (α =1, β =1) would achieve the highest renegotiation frequency, thus leading to the best network utilization. This research

investigates the achievable network performance of RED-VBR under the following three scenarios, (α =1, β =1), (α =1.1, β =0.9) and (α =1.3, β =0.7). Figure 2.6.2.1 compares the network utilization of the three scenarios for the video trace of *Robin Hood* with low quality.

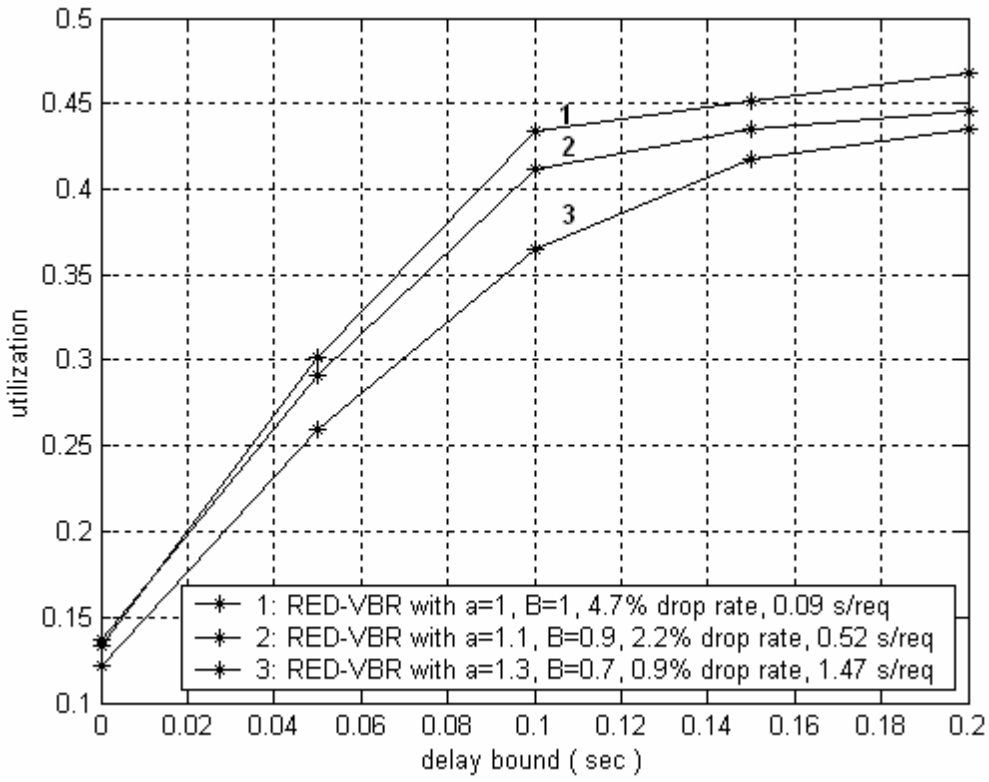


Figure 2.6.2.1: Online network performance for RED-VBR with various (α, β) pairs

As expected, the scenario with (α =1, β =1) achieves the best network utilization among the three scenarios, but has a rather small renegotiation interval of 0.09s. This renegotiation interval is much smaller than 0.52s for the scenario with (α =1.1, β =0.9), and 1.47s for the scenario with (α =1.3, β =0.7). This smaller interval would inevitably lead to a fairly large amount of control overhead. So the high network utilization achieved by this scenario is apparently compensated by the drawback.

It is worth noting that the previous studies [5, 7, 8] seem to investigate the performance mainly according to the average renegotiation interval and network utilization. Our study

additionally introduces another important performance metric, that is, the average drop rate. As shown in figure 2.6.2.1, besides having a too small renegotiation interval, the scenario with (α =1, β =1) has a relatively high drop rate of around 4.7%, larger than around 2.2% for the scenario with (α =1.1, β =0.9), and around 0.9% for the scenario with (α =1.3, β =0.7). This indicates that in addition to the average renegotiation interval, the average drop rate has an important impact on the network utilization as well. In order to further investigate the impact of the drop rate on the network performance with RED-VBR, we conduct another experiment. This experiment does not apply the traffic shaping technique; instead, the frames of admitted sources will directly go into the network buffer after receiving the response from the link control module. The drop rates are shown in table 2.6.2.1, denoted by "without policing."

QoS	Video Trace	Delay Bound (s)	Drop Rate			
			$\alpha = 1.1, \beta = 0.9$ with policing	$\alpha = 1.1, \beta = 0.9$ without policing	$\alpha = 1.3, \beta = 0.7$ with policing	$\alpha = 1.3, \beta = 0.7$ without policing
Low	Robin Hood	0	2.28%	0.01%	1.25%	0
		0.05	2.17%	0	0.93%	0
		0.1	2.17%	0.16%	0.74%	0.07%
		0.15	2.22%	0.27%	0.86%	0.20%
		0.2	2.13%	0.32%	0.82%	0.25%
High	Mr. Bean	0	0.93%	0	0.50%	0
		0.05	1.12%	0.05%	0.60%	0
		0.1	1.18%	0.14%	0.64%	0.02%
		0.15	1.11%	0.17%	0.59%	0.06%
		0.2	1.16%	0.16%	0.54%	0.08%

Table 2.6.2.1: Effect of policing for RED-VBR with various (α, β) pairs

Based on simulation results, for any scenario, the experiment without policing achieves a similar average renegotiation interval as the experiment with policing. However, the drop rate with policing is around ten times as large as that without policing. For example, for the delay bound of 0s, the drop rate of the scenario of *Robin Hood* with

$(\alpha = 1.1, \beta = 0.9)$ is greatly reduced from 2.28% with policing to 0.01% without policing. Simulation results show that the decrease of the drop rate without policing does not contribute much to the utilization improvement. This is probably due to the use of the (α, β) pair and the D-BIND bounding rates, and therefore RED-VBR allocates the additional bandwidth for each connection than the actual required bandwidth. The total excess reserved bandwidth could, to some degree, absorb the bursts in the future. Using policing can help provide the deterministic service for each segment, but would inversely reduce the SMG unnecessarily. Therefore, RED-VBR without policing may provide better performance together with a relatively low drop rate of lower than around 0.3% for either *Robin Hood* or *Mr. Bean*, as illustrated in table 2.6.2.1.

2.7 Summary

RED-VBR aims to provide a statistical service for VBR compressed traffic on top of a deterministic service. Based on our experiments of RED-VBR on a variety of video traces, some observations were provided. In order to reduce the implementation complexity of RED-VBR, we propose a new Virtual-Queue-Based RED-VBR scheme. This scheme greatly reduces the implementation complexity and simplifies the tuning of network parameters. Through analysis of the simulation results, we showed that the proposed Virtual-Queue-Based RED-VBR has comparable network performance to RED-VBR and Overlap-Based RED-VBR. Moreover, we provided an insight into RED-VBR through analysis of multiple performance metrics. We demonstrated that except the average renegotiation interval, the network performance has a high dependence on the average drop rate.

Chapter 3

R-VBR: Prediction-Based R-VBR

This chapter discusses Prediction-Based R-VBR and compares its performance with RED-VBR. Section 3.1 introduces a multiresolution learning-based neural network (NN) traffic predictor. Section 3.2 describes the segmentation algorithm for Prediction-Based R-VBR. The simulation setup for this scheme is depicted in section 3.3. Section 3.4 includes the simulation results and the analysis of Prediction-Based R-VBR with a comparison to RED-VBR. Finally, the main ideas presented in this chapter are summarized in section 3.5.

3.1 Traffic Prediction

A dynamic NRAM system based on online traffic prediction is considered to be very promising for live VBR compressed video traffic. As discussed in chapter 1, VBR compressed traffic exhibits burstiness over a wide range of time scales. This is because VBR compressed traffic has a time-dependent nature, formally characterized as self-similarity or long-range dependence (LRD) [19, 20]. This nature makes it possible to predict the future network traffic pattern according to the observed past traffic pattern. In general, the accuracy of a traffic predictor may be improved when the traffic predictor applies to traffic exhibiting strong LRD. Employing online traffic prediction would: 1) provide a sufficient leading-time for a dynamic allocation mechanism, which is especially

beneficial for the real-time applications; 2) improve the overall network performance due to the possibility to significantly decrease the renegotiation frequency.

Our experiment integrates a multiresolution learning-based NN traffic predictor, proposed and developed by Dr. Y. Liang [10]. This traffic predictor is a three-layer feed-forward neural network with a multiresolution learning paradigm. The multiresolution learning paradigm exploits the correlation structures in training data at multiple resolutions by decomposing the original signals and approximating these signals at different levels of detail. In other words, the multiresolution learning paradigm exploits the approximation sequence representation by representation, from the coarsest version to the finest version during the neural network training process. The NN traffic predictor with multiresolution learning can effectively generalize the traffic characteristics of the training data and improve the prediction accuracy and the robustness of the neural network traffic predictor, thus successfully performing a long-term prediction on VBR compressed traffic.

3.2 Prediction-Based R-VBR

Within Prediction-Based R-VBR, there is a dedicated traffic predictor associated with each stream control module as explained in section 1.1.3. Note that in RED-VBR, the stream control module updates the bounding rates using currently extracted bit statistics of the video stream and then uses these bounding rates to request resources for incoming frames. In Prediction-Based R-VBR, the bit statistics of currently extracted frames of the video stream will be first fed to a traffic predictor. The traffic predictor utilizes these statistics to predict incoming frames. These predicted frames are then utilized to compute the bounding rates for incoming frames and then make a resource request. Therefore, Prediction-Based R-VBR essentially allocates resources to future frames according to the most updated predictions for future frames.

Since the NN traffic predictor with multiresolution learning may predict long-term traffic with a time window up to hundreds of frames, it is unnecessary to update the bounding rates frame by frame to determine the renegotiation points as adopted by the segmentation algorithm of RED-VBR. Instead, choosing a relatively long interval to update the bounding rates is desirable and feasible. For simplicity, a fixed time window M is applied to segment an admitted video stream to provide a renegotiated service. That is, each admitted video stream regularly updates its bounding rates every M frames before the beginning of each new window (i.e., toward the end of each current window) except that:

- A previous renegotiation request from any admitted video stream is rejected.
- Admission of a new video source into the network is to be determined.

When meeting one of the above two exceptions, the admitted video stream will update the bounding rates immediately.

Note that Prediction-Based R-VBR will implement the following:

- The bounding rates for any video stream are always re-calculated based on the updated predictions for the incoming frames.
- Whenever the bounding rates for any video stream are updated, the bounding rates for all the other admitted video streams will be accordingly updated simultaneously.

The above two rules help ensure that the bandwidth re-allocation for all video streams is based on the most updated predictions for the incoming traffic, which enables more accurate bandwidth re-allocation. The detailed algorithm is described in figure 3.2.1.

Notation: M: the fixed time window;

$(f_1, f_2, ..., f_M)$: M consecutive frame sizes;

$(r_1, r_2, ..., r_M)$: the rates for M consecutive frames;

flag = NEW: the indication of the admission request from a new video stream;

flag = FAILURE/SUCCESS: the indication of renegotiation failure / success;

n: the number of frames processed;

n-success: the number of successful renegotiations;

n-failure: the number of failed renegotiations;

```
Process:
n = 0;
For ( i = 0; i < Max-Simulation-Frames; i ++ )
{
        If ( n = = M OR flag = = FAILURE OR flag = = NEW )
        {
                For ( j = 0;  j < Number-Admitted-Streams;  j ++)
                {       Predict the next M frames (f1, f2,..., fM);
                        Update (r1, r2,..., rM) based on (f1, f2,..., fM);          }
                Issue a renegotiation request;
                If ( admitted; )
                {       n = 0;
                        flag = SUCCESS;
                        n-success ++;    }
                Else if ( rejected; )
                {       flag = FAILURE;
                        n-failure ++;     }
        }
        Process the incoming frames from each stream;
        Transmit the incoming frames at the head of the outgoing queue into network;
        If ( flag = = SUCCESS )    {     n ++;    }
}
```

Figure 3.2.1: Procedure of Prediction-Based R-VBR

Usually, the updated prediction and the re-renegotiation will be conducted at the next frame time once the current renegotiation fails. In order to reduce the frequency of the traffic prediction and the re-renegotiation, we propose that the renegotiation request is issued based on the binary exponential backoff (BEB) algorithm instead of always at the next frame after fails. When a renegotiation is rejected at the first time, the next renegotiation has to wait 0 frame time or 1 frame time to be retried. If the second renegotiation is rejected again, the next renegotiation is then retried after waiting randomly either 0, 1, 2, or 3 frame times. As the number of consecutive rejections increases, the range of the waiting widow (i.e., number of frame-time) will increase exponentially ($2^F - 1$, where F is the number of consecutive rejections). Figure 3.2.2 describes the procedure of the proposed Prediction-Based R-VBR with the BEB algorithm for re-renegotiation.

Notation: F: the number of consecutive rejections;
W-window: the number of frames needed to wait for next renegotiation;
M: the fixed time window;
$(f_1, f_2, ..., f_M)$: M consecutive frame sizes;
$(r_1, r_2, ..., r_M)$: the rates for M consecutive frames;
flag = NEW: the indication of the admission request from a new video stream;
flag = FAILURE/SUCCESS: the indication of renegotiation failure / success;
n: the number of frames processed;
n-success: the number of successful renegotiations;
n-failure: the number of failed renegotiations;

Process with BEB:

```
F = 0; n = 0;
For ( i = 0; i < Max-Simulation-Frames; i ++ )
{       If ( flag = = FAILURE )
        {       F ++;   n = 0;
                Calculate the range of the waiting window: 2^F - 1;
                Randomly choose W-window which is in the range of 2^F - 1;
                W-window = min { W-window , M };            }
        Else if ( flag = = NEW )  {     n = 0; W-window = 0;    }
```

```
        If ( n = = M OR n = = W-window )
        {       For ( j = 0;  j < Number-Admitted-Streams;  j ++)
                {       Predict the next M frames (f1, f2,..., fM);
                        Update (r1, r2,..., rM) based on (f1, f2,..., fM);          }
                Issue a renegotiation request;
                If ( admitted; )  {       n = 0;  F = 0;
                                          flag = SUCCESS;
                                          n-success ++;    }
                Else if ( rejected; )   {   If ( flag != NEW ) { flag = FAILURE; }
                                          n-failure ++;     }         }
        Process the incoming frames from each stream;
        Transmit the incoming frames at the head of the outgoing queue into network;
        If ( flag = = SUCCESS )    {   n ++;      }
}
```

Figure 3.2.2: Procedure of Prediction-Based R-VBR with the BEB algorithm

3.3 Simulation Setup

The simulation setup of Prediction-Based R-VBR is similar to that of RED-VBR except for the following aspects:

- A traffic predictor is added to each stream control module. The traffic prediction model is regularly updated (i.e., re-trained with the current traffic data) after predicting some number of frames. This is because the prediction error will increase as the prediction proceeds. In our simulation, the prediction model was updated periodically after being used for predicting every about 10,000 frames. The first 488 frames of a video trace are used to build the first prediction model.
- We evaluate Prediction-Based R-VBR by experiments of two video traces. For each video trace, the fixed time window M is set to equal its minimal D-BIND window size used in RED-VBR.

- For low quality *Robin Hood*, M = 48 frames.
- For high quality *Mr. Bean*, M = 36 frames.

- In addition to the D-BIND traffic model, the statistical traffic model (using the actually arriving traffic rates instead of D-BIND rates) is tested in Prediction-Based R-VBR to better explore the statistical multiplexing gain.
- The admission control and renegotiation control algorithms in Prediction-Based R-VBR are the same as those used in RED-VBR, as discussed in section 2.2.3.
- No policing is used for Prediction-Based R-VBR.

3.4 Experiment Results and Analysis

3.4.1 Performance Comparison between Prediction-Based R-VBR and RED-VBR

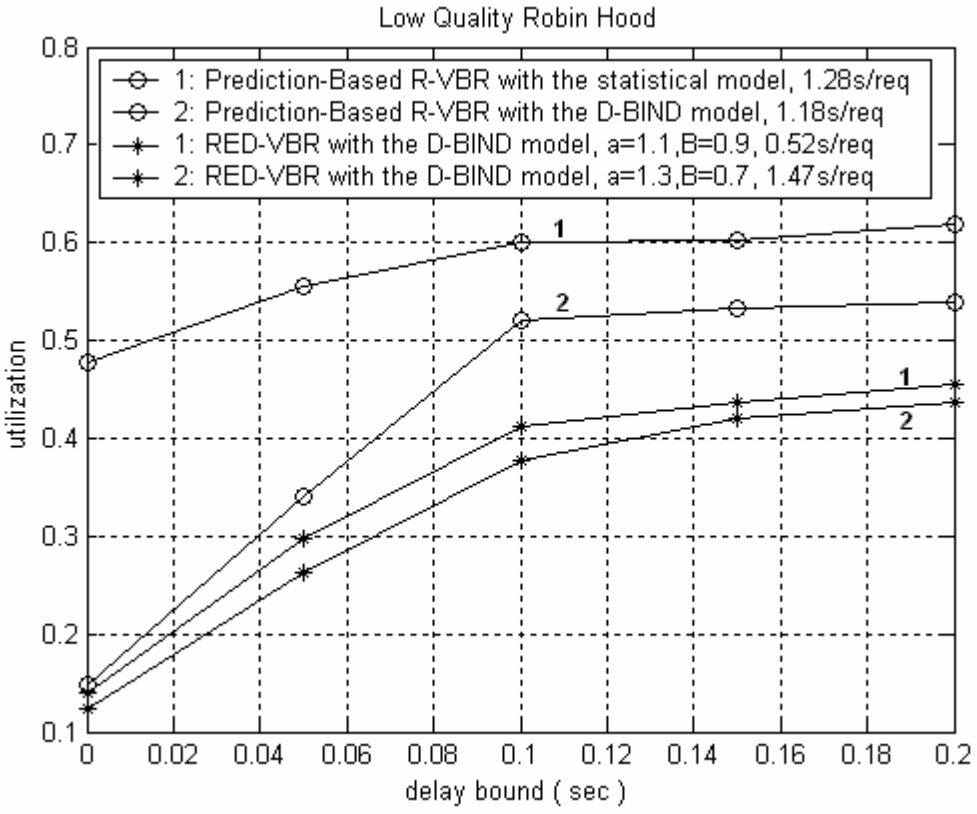


Figure 3.4.1.1: Effect of online R-VBR schemes for low quality *Robin Hood*

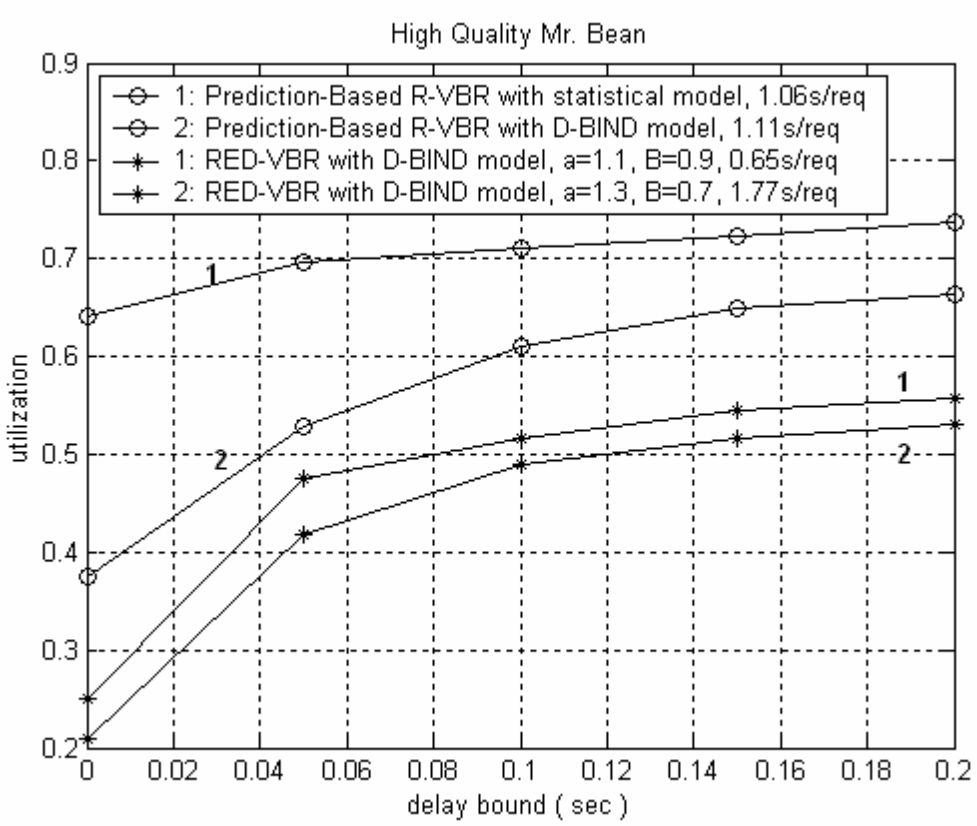


Figure 3.4.1.2: Effect of online R-VBR schemes for high quality *Mr. Bean*

Figure 3.4.1.1 and figure 3.4.1.2 compare the network performance between RED-VBR and Prediction-Based R-VBR for low quality *Robin Hood* and high quality *Mr. Bean*, respectively. We observe that Prediction-Based R-VBR has comparable average renegotiation interval as RED-VBR for either video trace. For low quality *Robin Hood*, the average renegotiation interval for Prediction-based R-VBR with the statistical traffic model and with the D-BIND traffic model is 1.28s and 1.18s, respectively. These two average renegotiation intervals are shorter than 1.47s for RED-VBR with $(\alpha = 1.3, \beta = 0.7)$, but are longer than 0.52s for RED-VBR with $(\alpha = 1.1, \beta = 0.9)$. On the other hand, for both video traces, the network utilization of Prediction-Based R-VBR with the D-BIND traffic model is consistently and significantly higher than that of RED-VBR for any delay bound. This improvement in the network utilization is particularly pronounced for Prediction-Based R-VBR with the statistical traffic model. For example, for delay bound of 0.2s, the network utilization of Prediction-Based R-VBR

with the D-BIND traffic model and with the statistical traffic model for low quality *Robin Hood* is around 53.9% and 61.9% respectively, which are much higher than those of RED-VBR, around 44%. The utilization improvement to RED-VBR is around 22.5% to 40%. Similarly, the utilization improvement of high quality *Mr. Bean* for delay bound of 0.2s ranges from around 20% to around 32%.

QoS	Video Trace	Delay Bound (s)	Average Drop rate			
			RED-VBR ($\alpha = 1.1, \beta = 0.9$)	RED-VBR ($\alpha = 1.3, \beta = 0.7$)	Prediction-Based R-VBR with D-BIND model	Prediction-Based R-VBR With statistical model
Low	Robin Hood	0	0.01%	0	0.09%	23.04%
		0.05	0	0	0.13%	9.27%
		0.1	0.16%	0.07%	0.93%	2.08%
		0.15	0.27%	0.20%	0.82%	1.65%
		0.2	0.32%	0.25%	0.91%	1.77%
High	Mr. Bean	0	0	0	0.63%	8.14%
		0.05	0.05%	0	0.26%	2.22%
		0.1	0.14%	0.02%	0.68%	2.12%
		0.15	0.17%	0.06%	0.97%	2.19%
		0.2	0.16%	0.08%	0.96%	2.27%

Table 3.4.1.1: Comparison of drop rate for various R-VBR schemes

It can be seen from table 3.4.1.1 that the average drop rate for Prediction-Based R-VBR with the use of the D-BIND traffic model is slightly higher than that of RED-VBR for any delay bound. The average drop rate for Prediction-Based R-VBR with the D-BIND traffic model is below 1%, and the average drop rate for Prediction-Based R-VBR with the statistical traffic model is below 3% for the delay bound longer than 0.1s, which are satisfying a wide range of real-time video applications in real world. Since the average renegotiation interval of Prediction-Based R-VBR is comparable to that of RED-VBR, it is apparent that the high network utilization of Prediction-Based R-VBR is due to its high

drop rate. In other words, the high network utilization is obtained at the expense of the high drop rate.

Based on the similar average renegotiation interval, RED-VBR achieves the lowest network utilization as well as the smallest drop rate. RED-VBR may be used to provide the best quality of service with a very low (close to zero) drop rate to each video stream. Prediction-Based R-VBR with the statistical traffic model achieves much higher network utilization and reasonable drop rate when the delay bound is longer than 0.1s. Prediction-Based R-VBR with the D-BIND traffic model seems to strike an efficient balance between the network utilization and the drop rate. Compared with RED-VBR, Prediction-Based R-VBR with the D-BIND traffic model achieves around 20% utilization improvement with an average drop rate less than about 1%.

Integrated packet-switching networks are expected to support all kinds of applications which have various requirements on the service quality. Some applications require a service with a risk that they may have to degrade a bit the quality of service if the renegotiation fails during the periods of network overload, but the service is more affordable. Some applications may ask a very strict service quality (e.g., with close to zero drop rate) and thus a more expensive service to guarantee their required resources even in the worst case. It would be good to employ different techniques/schemes to facilitate and provide appropriate services to meet the different needs of various applications. In this sense, Prediction-Based R-VBR, which may improve the overall network utilization by accommodating more traffic at the same time, is preferred by ordinary real-time multimedia applications, while RED-VBR is suitable for the prestigious applications with very high quality of service requirements.

3.4.2 Performance Comparison for Prediction-Based R-VBR with and without the BEB Algorithm

Figure 3.4.2.1 and figure 3.4.2.2 compare the network utilization of Prediction-Based R-VBR with the use of the BEB algorithm and without the use of the BEB algorithm for low quality *Robin Hood* and high quality *Mr. Bean*, respectively.

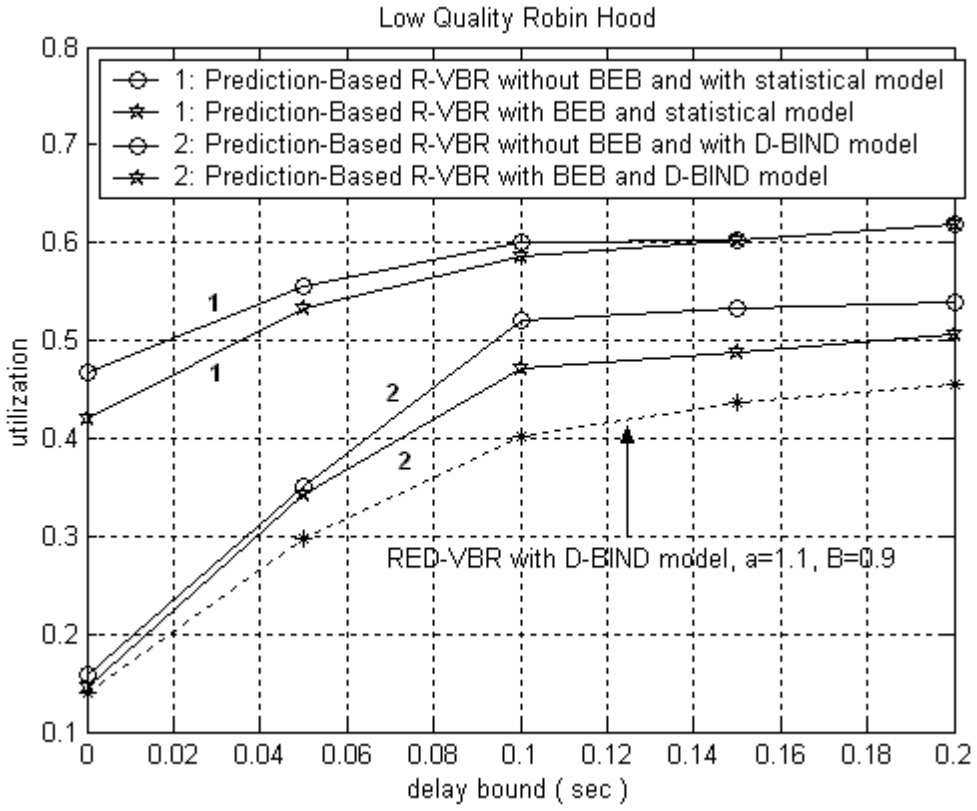


Figure 3.4.2.1: Effect of the BEB algorithm on network utilization for *Robin Hood*

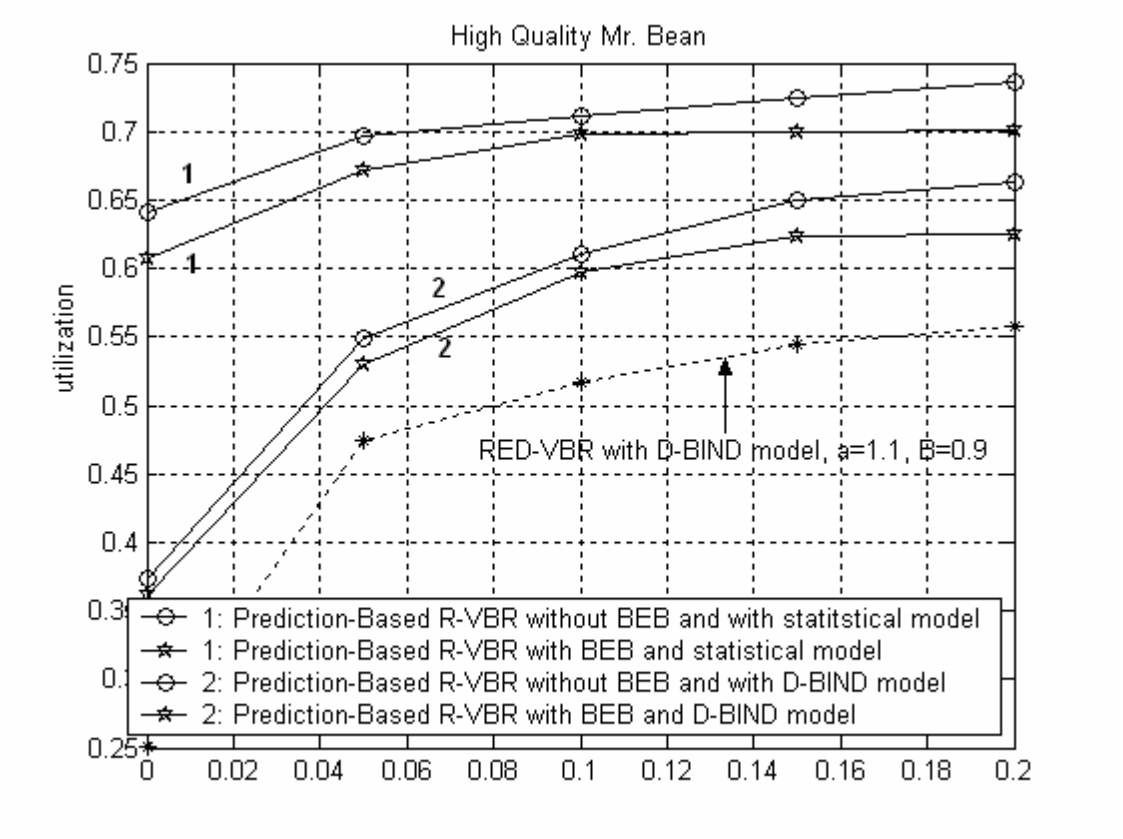


Figure 3.4.2.2: Effect of the BEB algorithm on network utilization for *Mr. Bean*

It is obvious that Prediction-Based R-VBR employing the BEB algorithm achieves a little lower network utilization than that without backoff on re-renegotiation. With the use of the BEB algorithm, as the renegotiation for resources is continuously rejected for several times, the next renegotiation may wait for a relatively long period because this waiting time is randomly determined based on the BEB algorithm. During this long period, there is a possibility that the bandwidth requested by the renegotiation may become available at some time. However, this available bandwidth cannot be immediately utilized until the waiting time is over. This is why Prediction-Based R-VBR with use of the BEB algorithm leads to a bit lower utilization than that without the use of the BEB algorithm.

The significant advantage of the use of the BEB algorithm is to greatly reduce the renegotiation frequency as shown in table 3.4.2.1. In addition, the lower average loss rate is also achieved. As for Prediction-Based R-VBR with the D-BIND traffic model, the improvement on the average renegotiation interval has been gained around 53.4% for low

quality *Robin Hood*, and about 23.4% for high quality *Mr. Bean*. A similar tendency can also be observed in Prediction-Based R-VBR with the statistical model. The apparently reduced renegotiation frequency means the significant reduction of the overhead of traffic prediction and renegotiation computation complexity, compensated to the slightly decreased network utilization.

QoS	Video Trace	Traffic Model used	Use BEB	Average Drop Rate	Average Rene-interval (s)
Low	Robin Hood	Statistical	No	7.56%	1.28
			Yes	7.09%	1.84
		D-BIND	No	0.58%	1.18
			Yes	0.37%	1.81
High	Mr. Bean	Statistical	No	3.39%	1.06
			Yes	2.82%	1.35
		D-BIND	No	0.70%	1.11
			Yes	0.52%	1.37

Table 3.4.2.1: Effect of the BEB algorithm on performance metrics

3.5 Summary

In this chapter, Prediction-Based R-VBR was studied through the multiresolution learning-based NN traffic predictor which has shown a good prediction performance in multi-step-ahead online traffic prediction. We demonstrated that compared with RED-VBR, Prediction-Based R-VBR with the D-BIND traffic model achieves significantly higher network utilization at a little expense of drop rate, and that with the statistical traffic model achieves substantially higher network utilization at a little expense of the drop rate when the delay bound is relaxed to longer than or equal to 0.1s. In order to reduce the renegotiation frequency, a novel use of the BEB algorithm for re-renegotiation was proposed to determine the next renegotiation point. This method would apparently increase the average renegotiation interval at a little expense of the network utilization.

Chapter 4

Conclusions and Future Work

This chapter summarizes the main contributions of this research, and provides the key results obtained as an outcome of this thesis. We then conclude the thesis by outlining some thoughts for future work. Section 4.1 discusses the major contributions of this research and Section 4.2 provides the pointers for further research in this area.

4.1 Contributions of this Research

This research was dedicated to R-VBR for VBR compressed video traffic. The objective of designing R-VBR is to make efficient use of the limited capacity available in the packet-switching networks while enabling any desired quality of service for diverse multimedia applications. The overall network performance of any R-VBR scheme would be affected by several conflicting factors, such as, high network utilization and high QoS, high network utilization and low renegotiation frequency. Different R-VBR approaches provide various balances among these conflicting factors. Our research studies two R-VBR schemes, RED-VBR and Prediction-Based R-VBR.

The main contributions of this research are summarized below:

- Implementation of RED-VBR on various types of real-world video traces. These video traces are MPEG-4 encoded, either from high encoding quality or low encoding quality. Through simulations, we identified that the D-BIND window size is video trace dependent and is critical in implementing RED-VBR correctly. We then empirically determine the minimal window size for all tested video traces.
- Analysis of the potential relationship of performance of video traces according to the characteristics of video traces. We demonstrated that low quality video traces achieve low network utilization than high quality video traces. This is probably because the low quality compression technique gives rise to a much more bursty video stream than the high quality compression technique.
- Development of a new Virtual-Queue-Based RED-VBR scheme to reduce the computational complexity and memory space required by RED-VBR. This new scheme can conveniently control the performance by adjusting a tuning parameter. We further presented an empirical equation to determine the maximum tuning parameter. The tuning parameter, equal to or smaller than the maximum tuning parameter, would guarantee no unexpected drops at the buffer of the outgoing link. Based on simulation results of a variety of scenarios, multiplexed video streams from homogeneous video trace and from heterogeneous video traces, we demonstrated that Virtual-Queue-Based RED-VBR can achieve comparable network performance as RED-VBR and Overlap-Based RED-VBR. The performance improvement of Virtual-Queue-Based RED-VBR over RED-VBR and its variant becomes prominent when testing on the heterogeneous video traces scenarios.
- Evaluation of the performance of the Prediction-Based R-VBR scheme which uses a multiresolution learning neural network traffic predictor. Compared with RED-VBR, we presented that Prediction-Based R-VBR achieves a similar average renegotiation interval, but significantly higher network utilization and with a slightly higher drop rate. In order to further reduce the renegotiation frequency, a novel use of the BEB algorithm for the re-renegotiation was then proposed to determine the next renegotiation point. This method would greatly

increase the average renegotiation interval at a little expense of the network utilization.

- Analyzing the overall performance by adding the average drop rate, in addition to the network utilization, average renegotiation interval, renegotiation blocking probability and delay. We observed that with a comparable renegotiation interval, higher drop rate will lead to higher network utilization. A clear big picture of the performance of the studied R-VBR approaches was provided, which may be served as the guideline to suggest appropriate R-VBR schemes for applications with different QoS requirements.

4.2 Future Work

Based on our current research, we provide the following thoughts for future work in this area.

- To study what kind of network performance may be achieved when multiplexing different types of video traces. We observed that different video traffic has different minimal D-BIND window size. However, we do not know their exact impacts on network performance. A better understanding of this can help better manage network resources.
- To develop an approach to determine the maximum delay bound in equation 2.4-1. With our Virtual-Queue-Based scheme, we gave an empirical equation to determine the value of the maximum tuning parameter for each delay bound. In this equation, the maximum delay bound is unknown and is video trace dependent. With the knowledge of this parameter, the value of the maximum tuning parameter for any delay bound can be easily derived.
- To investigate the impact of accuracy of prediction model on network performance.

References

[1] A. A. Lazar, G. Pacifici, and D. E. Pendarakis, "Modeling video sources for real-time scheduling," in *Proceedings of IEEE GLOBECOM'93*, pp. 835-839, Houston, TX, November 1993.

[2] D. Lucantoni, M. Neuts, and A. Reibman, "Methods for performance evaluation of VBR video traffic models," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 176-180, April 1994.

[3] M. W. Garrett and Walter Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proceedings of ACM SIGCOMM'94*, London, U.K., pp. 269-280, August 1994.

[4] M. W. Garrett, "Contributions toward real-time services on packet switched networks," Ph.D. dissertation, Columbia University, New York, NY, 1993.

[5] H. Zhang and E. W. Knightly, "RED-VBR: A new approach to support delay-sensitive VBR video in packet-switched networks," in *Proceedings of Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'95)*, pp. 275-286, April 1995.

[6] K. Liu, D. W. Petr, V. S. Frost, H. Zhu, C. Brown, and W. Edwards, "Design and Analysis of a Bandwidth Management Framework for ATM-Based Broadband ISDN," *IEEE Communications Magazine*, pp. 138-145, May 1997.

[7] P. Bocheck and S.-F. Chang, "Content-based VBR traffic modeling and its application to dynamic network resource allocation," Columbia University, New York, Res. Rep. 48c-98-20, 1998.

[8] M. Wu, R. A. Joyce, H.-S. Wong, L. Guan, and S.-Y. Kung, "Dynamic resource allocation via video content and short-term traffic statistics," *IEEE Transactions on Multimedia*, vol 3, No. 2, pp. 186-199, June 2001.

[9] M. Grossglauser, S. Keshav, and D. Tse, "RCBR: A simple and efficient service for multiple time-scale traffic," in *Proceedings of SIGCOMM'95*, pp. 219-230, Boston, MA, September 1995.

[10] Y. Liang, "Real-Time VBR video traffic prediction for Dynamic Bandwidth allocation", *IEEE Transactions on Systems,* vol. 34, pp. 32-47, 2004.

[11] E. W. Knighly and H. Zhang, "D-BIND: An accurate traffic model for providing QoS guarantees to VBR traffic," *IEEE/ACM transactions on Networking*, vol. 5, pp. 219-231, 1997.

[12] C. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Transactions on Automatic Control*, vol. 39, pp. 913-931, May 1994.

[13] R. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, pp. 114-121, January 1991.

[14] J. Liebeherr, D. Wrege, and D. Ferrari, "Exact admission control for networks with bounded delay services," Technical Report CS-94-29, University of Virginia, Department of Computer Science, July 1994.

[15] H. Zhang and D. Ferrari, "Improving utilization for deterministic service in multimedia communications," in *Proceedings of 1994 International Conference on Multimedia Computing and Systems*, pp. 295-304, Boston, MA, May 1994.

[16] Public library of frame-size video traces: http://www-tkn.ee.tu-berlin.de/research/trace/trace.html.

[17] A. A. Lazar and G. Pacifici, "Control of resources in broadband networks with quality of service guarantees," *IEEE Communications*, pp.66-73, October 1991.

[18] S. Chong, S. Q. Li and J. Ghosh, "Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM," *IEEE Journal on Selected Areas of Communications*, vol. 13, pp. 12-23, January 1995.

[19] K. Park and W. Willinger, *Self-similar network traffic and performance evaluation*, Wiley, 2000.

[20] J. Beran, *Statistics for long-memory processes*. Chapman and Hall, 1994.

Glossary of Acronyms

AN: Access Network

BEB: Binary Exponential Backoff

CBR: Constant Bit Rate

D-BIND: Deterministic Bounding Interval-Length Dependent

D-VBR: Deterministic Variable Bit Rate

FCFS: First Come First Serve

GOP: Group of Pictures

IntServ: Integrated Services

LAN: Local Area Network

MPEG: Moving Pictographic Expert Group

NRAM: Network Resource Allocation and Management

Prediction-Based R-VBR: Prediction-Based Renegotiated Variable Bit Rate

QoS: Quality of Service

R-CBR: Renegotiated Constant Bit Rate

RED-VBR: Renegotiated Deterministic Variable Bit Rate

R-VBR: Renegotiated Variable Bit Rate

SMG: Statistical Multiplexing Gain

VBR: Variable Bit Rate