AN ADMISSION CONTROL ALGORITHM FOR PROVIDING QUALITY-OF-SERVICE GUARANTEE FOR INDIVIDUAL CONNECTION IN A VIDEO-ON-DEMAND SYSTEM

By

Xiaoqing Wang

Supervised By :

Prof. John C.S. Lui

A Thesis Submitted in Parital Fulfilment
of the requirements for the degree of
Master of Philosophy
in
Computer Science & Engineering

JUNE, 2000

Acknowledgments

I would like to thank John Lui, my thesis supervisor, for his patience, guidance and support. His creative, insightful comments and experience have provided me with precious help to my thesis. More importantly, I am eternally grateful for his encouragement and spiritual support. Also, I would like to thank him for giving me a chance to study here as a graduate student by supporting me in these two years. Without his generous financial support, I just cannot come here, let alone studying here.

I would like to express my sincere thanks to Clement Lee for giving me many precious helps in these two years.

Special thanks to David Yao who taught me a lot on queueing theory which is very helpful to my research.

I would extend my cordial thanks to my colleagues. They helped me to adapt the new environment, provided me technical and non-technical information, fun and other help in these two years.

Finally, I would like to express my sincere appreciation to my family for their constant support during the process of my graduate study.

AN ADMISSION CONTROL ALGORITHM FOR PROVIDING QUALITY-OF-SERVICE GUARANTEE FOR INDIVIDUAL CONNECTION IN A VIDOE-ON-DEMAND SYSTEM

submitted by

Xiaoqing Wang

for the degree of Master of Philosophy
at the Chinese University of Hong Kong

Abstract

Advances in distributed multimedia systems and networking technologies have made it feasible to provide interactive digital services such as video-on-demand (VoD) services. Designing and implementing a cost-effective VoD system so as to deliver video services is a challenging task due to the quality of service (QoS) requirements of digital video.

In this thesis, we consider a theoretical framework for performing admission control in a VoD system. Previous work on the admission control on VoD system guaranteed QoS only on the aggregated traffic, our admission control algorithm can guarantee the QoS requirements of individual connections. Also, currently few papers considered both admission control for VoD servers and for the network sub-system. For the VoD service, the user can only view the video at the end of the transmission networks. So the transmission of video stream in the network sub-system is also very important. Our algorithm presented in this thesis is a two-step admission control algorithm. The QoS requirements are bandwidth and packet dropping rate (the expectation of maximum number of packets can be dropped in a transmission time frame), which are specified by users. By using Chernoff's theorem, we can statistically guarantee the bandwidth requirements of users'. By using the strong conservation laws, the admissible region for the packet dropping rate is easily derived and the admissible control policy which achieves the given dropping rate vector (or even "better vector") can be found. Experiments show that our proposed algorithm can achieve high bandwidth utilization, making VoD service cost-effectively.

一個提供對Video-on-Demand系統中各個用戶質量保證的接納控制算法

作者：王小青

摘要

分佈式多媒體系統和網絡技術的發展，使象Video-on-Demand(VoD)這類互動數字化服務成爲可能。由于數字化video的質量要求，設計和實現一個費用合理的VoD系統來傳遞video服務是一個艱巨的任務。

在這篇論文中，我們考慮一個在VoD系統中進行接納控制的理論框架。之前的有關VoD系統的接納控制的工作只能保證整體用戶的質量要求，我們的接納控制算法能保證對每一用戶的質量要求。而且，現在很少有論文考慮對VoD服務器和網絡系統的接納控制。對于VoD服務，用戶只能在傳輸網絡終端上看到video，所以video在網絡上的傳輸也是很重要的。我們在這篇論文中提出的算法是一個兩步驟的接納控制算法。質量要求包括帶寬的要求和packet的放棄率(在一個傳輸週期內最多可以放棄packet的個數的期望值)。這些要求是由用戶提出的。通過運用Chernoff定理，我們可以在統計上保證用戶的帶寬要求。通過運用強守恆律，packet放棄率的可行區域可以容易地找到，而且可以達到給定packet放棄率(或更好的packet放棄率)的接納控制也可以找到。試驗顯示我們所提出的算法可以達到帶寬的高利用，使VoD服務費用合理。

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the exciting development in our society in the last decade is the development of communication networks [21]. The transmission media evolved from copper-based transmission media to optical fiber, made the speed of transmission and capacity of the networks improved significantly. Various kinds of switches and routers are built to handle more and more complicated communication traffic. The computers used in the communication networks also enjoyed a great development. For example, the growth rate of computer speed is that it can increase by 100% every 18 months. All of these give the current communication networks the ability to develop many new applications. Video-on-Demand (VoD) is one such application which can provide videos to the users who are sitting at home. For example, users can select videos within the VoD provider's video collections, choose how to view the selected video under different playback modes, e.g., fast forward, fast backward, etc. This is just like viewing the video by using a conventional VCR, but without the inconvenience to go to the video rental shop to get the wanted videos [7].

However, designing and implementing a cost-effective VoD system so as to deliver video services is a challenging task due to the quality of service (QoS) requirements of digital video. For example, each user may demand to view the video at certain frame rate. This frame rate requirement can be translated to a bandwidth requirement of transmitting the video data. Since most of the video data are variable-bit-rate (VBR) in nature, to satisfy this requirement and at the same time, to use the bandwidth of the transmission network efficiently, we have to perform the necessary bandwidth allocation and management in transmission network. Another implication of the QoS requirement is that we have to perform the proper I/O (or disk) scheduling so that the data correspond to any video stream can be retrieved in time for delivery. Therefore, in order to maximize the number of users we can support in a VoD system, we have to have a careful planning of I/O scheduling as well as network bandwidth management. Employing some form of admission control mechanism is one way to address this problem.

The admission control is a mechanics to limit the number of users admitted in the VoD system so that the specified QoS requirements of the users currently in the system can be satisfied. The design of admission control algorithms is very important for VoD system because if the admission control in VoD system is too conserved, the number of the users the system admits will be less than the number of the users the system can support, this will result in the underutilization of system resources. On the other hand, if the admission control permits too many users to enter the system, the scarcity of system resources will make some users' QoS requirements violated. Due to the scarcity of system resources, the admission control algorithms should be carefully designed to make high utilization of system resources and at the same time, satisfy all the admitted users' QoS requirements. In general, when a request arrives to the VoD system, the admission control algorithm needs to decide whether to accept or reject this request based on the following two criteria:

- Whether the system has enough resources to satisfy the QoS requirements of this request.

- If the system decides to accept this new request, the system also needs to ensure that the QoS requirements of other existing users will not be violated.

In this thesis, we present a simple yet efficient admission control algorithm which can give statistical guarantee for the QoS requirements of each individual connection and at the same time, achieve efficient usage of system resources.

The organization of this thesis is as following:

In Chapter Two, we briefly describe the general architecture of the VoD system and the related issues. We explain why the video streams are VBR in nature and which compression technology is usually used for the videos in VoD system. We explain some kinds of storage media for VoD system. We explain the data placement schemes in VoD system. We also give an overview of disk scheduling algorithms in the VoD system. Lastly, we explain the admission control in VoD system.

In Chapter Three, we explain the QoS requirements we choose and the system model we use. Then we present our admission control algorithm. Some preliminaries related to stochastic scheduling via polymatroid structure is also presented.

In Chapter Four, we present some experimental results obtained by using our proposed admission control algorithm. We also compare these results with the results obtained by using average bandwidth allocation strategy.

In Chapter Five, we give the conclusion that our proposed admission control algorithm can guarantee the QoS requirements of individual connection and at the same time, achieve high bandwidth utilization. We also discuss some future work.

Chapter 2

The General Architecture of the VoD System and the Related Issues

In this chapter, we briefly describe the general architecture of the VoD system and the related issues. We explain why the video streams are VBR in nature and which compression technology is usually used for videos in VoD system. We explain some kinds of storage media for VoD system. We explain the data placement schemes in VoD system. We also give an overview of disk scheduling algorithms in the VoD system. Lastly, we explain the admission control in VoD system.

2.1 A Brief Description of VoD System

A typical VoD system consists of a storage sub-system and a network sub-system. The users view the videos on their display units at the end of network sub-system. Figure 2.1 in [7] illustrates a typical VoD system.

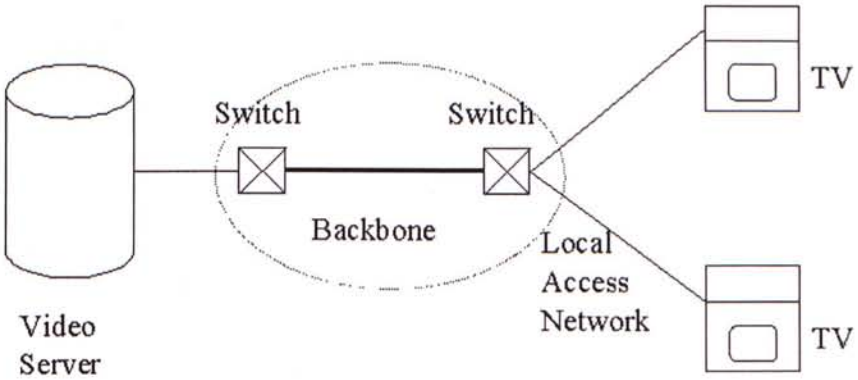

Figure 2.1: A VoD System [7]

The storage sub-system has the functions to store the video files (movies) and when a view request is admitted, retrieve the corresponding video file in the storage sub-system and transfer video streams to the interface between the storage sub-system and network sub-system.

There are several kinds of media that can be used to store the video files. The cheapest one is to use magnetic tapes to store the video files. Other kinds of storage media include optical storage, magnetic disks and RAM. We will discuss these storage media in detail in section 2.3.

The data placement scheme in VoD system is an active research field in recent years. How to choose a data placement scheme so as to make the retrieval of requested video files more efficiently is an important issue related to the performance of the VoD system. We will talk about data placement schemes in detail in section 2.4.

How to retrieve the requested video files so as to satisfy the real-time constraint and other required QoS requirements is an important problem. This problem is closely related to the disk scheduling, which is also an active research field in recent years. We will give an overview of disk scheduling in VoD system in section 2.5.

Before we explain all these topics in detail, we first explain why video streams in VoD service are VBR in nature.

2.2 Why Video Streams in VoD Service are VBR in Nature?

In VoD service, the original video files (movies) are usually very big. They are usually compressed to reduce the storage and bandwidth requirements. For video compression, MPEG (Motion Picture Experts Group) compression is typically used in recent years. There are some special properties with videos. For example, a moving picture consists of a succession of still images, we call these still images "frames". If there is no much motion between several successive frames, these frames will carry almost the same information. Even for the successive frames with much motion, some background information in the successive frames is identical. This kind of redundancy is called inter-frame redundancy. MPEG considers this property of videos. When using MPEG to compress the video files, the inter-frame redundancy is removed. So when the successive original frames of video containing some identical information are compressed, each of the output frames except the first one will contain much less bits related to the identical information than that of the first frame. That is, for different original frames, the compression rate (which is expressed by the original frame size divided by the size of the compressed corresponding frame) is different. Since the video stream should be displayed on the users' display units with fixed frame rate to get the acceptable quality of video in the users' end, translating this fixed frame rate to bit rate, we find the bit rate is an random variable. Hence the video streams in VoD system are VBR in nature.

Figure 2.2 illustrates the trade-off between image quality and output bit rate for a typical compression technology [10]. In this figure, video quality is measured by the distortion in a video frame which is the difference between the quality of the original image before compression and that after compression. The more the motion between successive frames, the larger the bits in the corresponding output frames, and the larger bit rate needed to transfer or transmit such a frame.

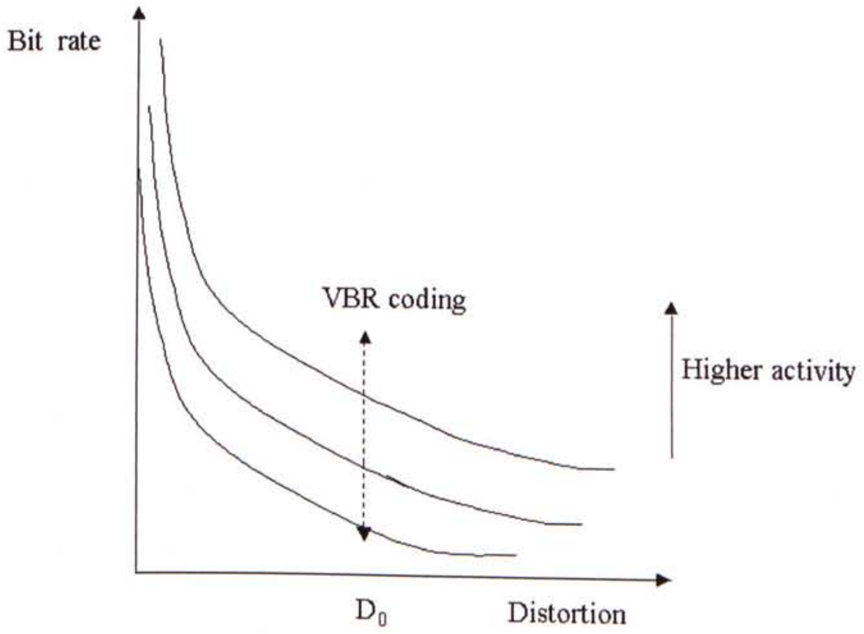

Figure 2.2: Trade-off Between Quality and Bit Rate in Video Compression [10]

In this thesis, we consider the video files which are compressed using MPEG-1 and MPEG-2.

2.3 The Video Storage Media in the VoD Systems

There are several kinds of media that can be used to storage video files. For example, magnetic tape, optical storage, magnetic disks and RAM. They all have advantages and drawbacks when they are used to store video files. In practice, not all movies have the same popularity and the popularity of movies is distributed based on the Zipf's law [22]. This implies that we can use a storage hierarchy to store the movies. In this section, we briefly explain each of the above storage media.

The cheapest way to store video files is on magnetic tape. When the movie is compressed using MPEG-2, the rough size of the compressed movie is usually 4 GB. According to [14], a DAT tape can store two movies (8 GB) at a cost of 5 dollars/gigabyte. There are large mechanically tape servers that hold tapes. A robot arm is used to fetch the requested tape from the tape server and insert it into a tape driver. The drawbacks with this storage systems are the long fetching time, the low transfer rate and the very limited number of users can be serviced at the same time(the maximum number of users is equal to the number of tape drivers in the storage system).

Another way to store video files is on optical storage. Although currently a CD-ROM can only store 650 MB data, the next generation CD-ROM is believed to be able to store 4 GB(one movie). The CD-ROM has low cost and high reliability, making it to be one good candidate of the storage media to store most popular movies.

The third way to store video files is on magnetic disks. The access time for disk is very short (10 msec), and their transfer rate is high (greater than 10 MB/sec) and its capacity is 10 GB [14]. The main drawback of this type of storage media is relatively high cost.

The fourth way to store video files is on RAM. This is the fastest way, but is also the most expensive way. RAM is most suitable to store the very popular movies which usually have multi-viewers simultaneously.

2.4 The Data Placement Scheme in the VoD System

Choosing the proper data placement scheme is very important for designing a cost-effective VoD system. We use "disk" here to represent the storage media the VoD system use to store the video files (movies). The video files must be divided into "blocks" before they can be stored in the disks. How to place the video data in the disk so as to achieve maximum benefit from it (e.g., maximizing the number of users that the VoD system can support) is an active research area. In this section, we will briefly explain two data placement schemes in current literature.

One of the data placement schemes is called disk farm [14]. Each disk holds multiple entire movies and each movie should be held at least in two disks for reliable reason. This data placement scheme has some drawbacks if employed in a VoD system: On one hand, because there maybe multiple users who want to view a same movie simultaneously, putting an entire movie in a single disk will limit the number of users who can view the same movie at the same time due to the disk I/O bandwidth and the real-time constraint of the video streams; on the other hand, if each movie is held on several disks to support more users who want to view the same movie simultaneously, the storage capacities of the disks are not efficiently utilized because we are duplicating too many movies.

Another kind of the data placement schemes is called RAID [14](redundant array of inexpensive disks). Each movie is stored in several disks. For example, block 1 of the movie is stored in disk 1, block 2 of the movie is stored in disk 2,...,block N of the movie is stored in disk N, block $N+1$ of the movie is stored in disk 1, etc. See Figure 2.3 for an illustration.

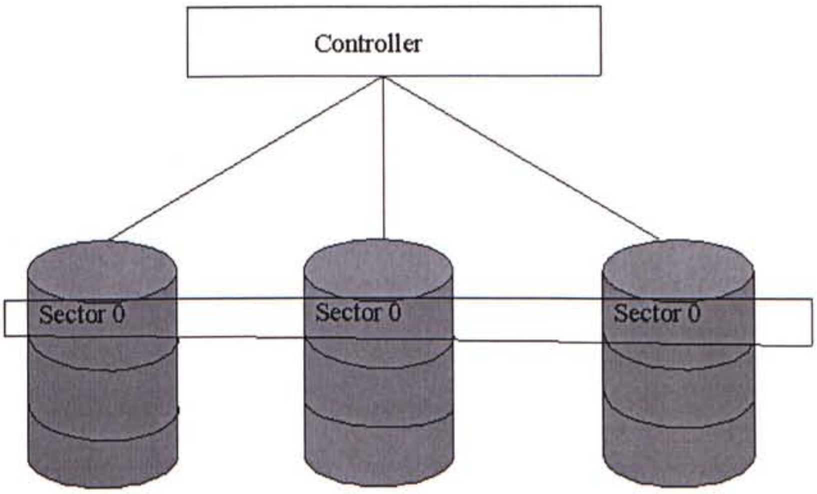

Figure 2.3: A Simple RAID Architecture [5]

RAID is better than disk farm in that multi-disks can be accessed in parallel and independently, so the system can support more users who want to view a movie simultaneously, probably starting from the different parts of the same movie.

2.5 An Overview of Disk Scheduling in VoD System

Video streams have rather stringent QoS requirements. For example, the video streams must be transferred in real-time in order to satisfy the view requirements of users (e.g., 30 frame per second, 1.5 Mbps, etc.). This presses extra requirements on disk scheduling algorithms and the traditional disk scheduling algorithms cannot be used directly for VoD application. In this section, we give a brief overview of disk scheduling algorithms in VoD system presented in the literature [5].

One of the algorithms for disk scheduling in VoD system is Round-Robin. In each service round, the disk head accesses the video files in a fixed order. This algorithm can provide fair access to each video stream and with reasonable throughput. It can also guarantee the real-time constraints for each video stream if one appropriately chooses the block size for each video stream. It is simple and easy to be implemented. But it does not exploit the relative positions of different video blocks in one disk. So the seek time may be unnecessary long and the disk bandwidth utilization is not high.

Another algorithm for disk scheduling in VoD system is earliest deadline first(EDF). In this algorithm, first, we need compute deadline for each block of video files according to the real-time constraints and other QoS requirements. Then, the disk head accesses the video files according to the deadline of the video blocks. This algorithm can guarantee real-time constraints of each video stream by scheduling the video blocks with the earliest deadline first. But this algorithm does not exploit the relative positions of different video blocks in one disk, either. So the disk bandwidth utilization is not high, either.

In order to exploit the relative positions of different video blocks in one disk to reduce the total seek time, another disk scheduling algorithm — SCAN-EDF — was introduced. The SCAN algorithm schedules the disk head access according to the relative positions of different video blocks in one disk. By eliminating the backtracking of the disk head access, SCAN can reduce the total seek time significantly. SCAN-EDF schedules the blocks of video streams according to earliest deadline first. When there are several blocks of different video streams with the same deadline, SCAN is used to schedule their relative retrieval order.

This algorithm can guarantee the real-time constraints of the video streams and at the same time, reduce the total seek time.

2.6 The Admission Control in VoD System

In order to make VoD service cost-effective, one major issue is to effectively and efficiently use system resources, such as disk I/O bandwidth and network bandwidth. Although we can use parallel disks to increase the I/O bandwidth, it is still a possibility for bottleneck and one has to seriously consider the I/O scheduling algorithm. Although the bandwidth of the networks has been greatly improved in recent years, because the transmission networks are shared by many applications, it is still a scarce system resource.

Each user has some QoS requirements. For example, each user may demand to view the video at certain frame rate. This frame rate requirement can be translated to a bandwidth requirement of transmitting the video data. Since most of the video data are VBR in nature, to satisfy this requirement and at the same time, to use the bandwidth of the transmission network efficiently, we have to perform the necessary bandwidth allocation and management in transmission networks. Another implication of the QoS requirements is that we have to perform the proper I/O (or disk) scheduling so that the data correspond to any video stream can be retrieved in time for delivery. Therefore, in order to maximize the number of users we can support in a VoD system, we have to have a careful planning of I/O scheduling as well as network bandwidth management.

Employing some form of admission control mechanism is one way to address this problem. In general, when a request arrives to the VoD system, the admission control algorithm needs to decide whether to accept or reject this request based on the following two criteria:

- Whether the system has enough resources to satisfy the QoS requirements of this request.
- If the system decides to accept this new request, the system also needs to ensure that the QoS requirements of other existing users will not be violated.

There is much work done for the admission control algorithms for VoD servers, to name a few, we have [8, 11, 15, 12, 9]. Few papers considered both admission control for VoD servers and the transmission networks. For the VoD service, the user can only view the video at the end of the transmission networks. So the transmission of video stream in the transmission networks is also very important. Since the VoD service has rather stringent QoS requirements, especially the real time constraint, the admission control algorithm should be carefully designed and easily implemented.

Our algorithm presented in this thesis is a two-step admission control algorithm. The QoS requirements are bandwidth and packet dropping rate (the expectation of maximum number of packets can be dropped in a transmission time frame), which are specified by users. By using Chernoff's theorem, we can statistically guarantee the bandwidth requirements of users'. By using the strong conservation laws, the admissible region for the dropping rate is easily derived and the admissible control policy which achieves the given dropping rate vector can be found. Our storage server admission control algorithm is similar to [8], but we consider more complex cases. Our network admission control algorithm is similar to [2], but with more careful design and simpler presentation.

Chapter 3

Our Admission Control Algorithm for VoD System

In this chapter, we explain the QoS requirements we choose and the system model we use. Then we present our admission control algorithm. Some preliminaries related to stochastic scheduling via polymatroid structure is also presented.

3.1 QoS Requirements We Choose

QoS is a generic notion. It means different things for different people. In VoD service, QoS usually refers to the service quality that the end users specified.

In this thesis, the QoS requirements we choose are bandwidth requirement and packet dropping rate requirement which are specified by users.

Bandwidth requirement — In order to get normal quality of video at the end users' display units, the video must be played at a fixed frame rate. For example, for video compressed by MPEG, this rate is 30 frame/sec (NTSC). It is 1.5 Mbps (average) for MPEG-1 compressed videos and 4–15 Mbps (average) for MPEG-2 compressed videos. To guarantee this kind of play rate, we should allocate enough bandwidth, both in disks (that is, disk I/O) and transmission networks.

Packet dropping rate requirement — Because of the real-time constraints of the digital videos, the packet will be dropped if it has excessive delay. In this thesis, we consider the packets in network sub-system which should be transmitted in current transmission time frame but can not be transmitted as violating the deadline and must be dropped.

3.2 System Model

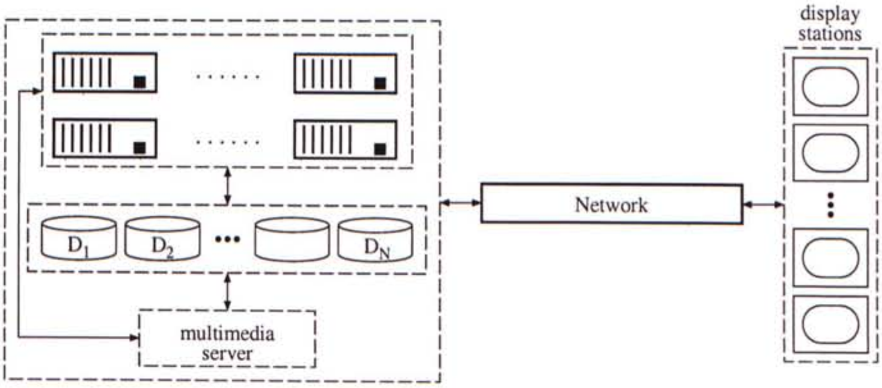

Figure 3.1: Multimedia Storage Server Architecture.

Figure 3.1 illustrates the VoD server we consider in this thesis. The client can send in a video viewing request to the VoD server, and the VoD server has to perform the necessary admission control, both for the server I/O bandwidth resource as well as the network bandwidth resource.

In general, the VoD server also needs to perform storage management for all video data. There are many different data placement techniques reported in the literatures [6], for example, for a large video file (e.g., movie), we may consider to stripe the data evenly across all disks in the system. For many small and popular video files (e.g., commercial), we may have to classify these short clips as a set of objects and assign each video object to a disk so as to achieve storage and load balanced feature (see Chapter Two for a detailed description).

To service video request, the disk storage system usually employs some form of the cycle-based (or group-based) scheduling algorithm [3, 16, 20]. Let us briefly describe the cycle-based scheduling algorithm here. In cycle-based scheduling algorithms, the retrieval of data from the disk sub-system is performed on a cyclic basis where each cycle is of length T and in each cycle, the system retrieves data for n video requests. Under the cycle-based scheduling algorithm, the transmission of data retrieved from the storage system in the i^{th} cycle does not start until the end of the i^{th} cycle[1]. This is motivated by the increased opportunities for performing seek optimization (i.e., data blocks needed for service are retrieved using a SCAN-type algorithm). The cost of this optimization is that the system needs additional buffer space to hold the retrieved data until the beginning of the next cycle. This cycle-based (or group-based) approach to servicing video request streams is, for instance, suggested in [3, 16, 20], and the trade-off between improved utilization of the disk bandwidth (due to seek optimization) and the need for additional buffer space is analyzed in several works[2], e.g., [3, 1, 20]. Figure 3.2 illustrates the cycle-based scheduling algorithm.

In the figure, the system is retrieving data for three video streams. The requirement is that all retrieved data must be serviced (be delivered to the network module for network transmission) at the end of every cycle. Since the disk is performing some form of SCAN algorithm at each cycle so as to reduce the overall seek overhead, the *order* of data retrieved from various streams is different from cycle to cycle. Also, depending on the data placement policy, it is possible that

---

[1] That is, here we assume that the server is responsible for maintaining the continuity in data delivery, where the clients have relatively little buffer space. Thus, if the data delivery is not "offset" by one cycle from data retrieval, jitter may occur.

[2] In general, larger values of n afford better seek optimization opportunities, but they also result in larger buffer space requirement.

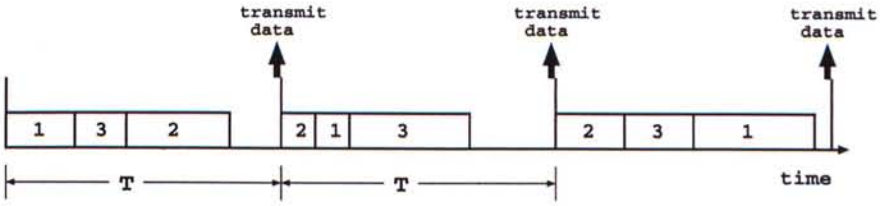

Figure 3.2: Cycle-based Scheduling Algorithm for Retrieving Three Streams of Data.

the size of data retrieved for each video stream varies from cycle to cycle.

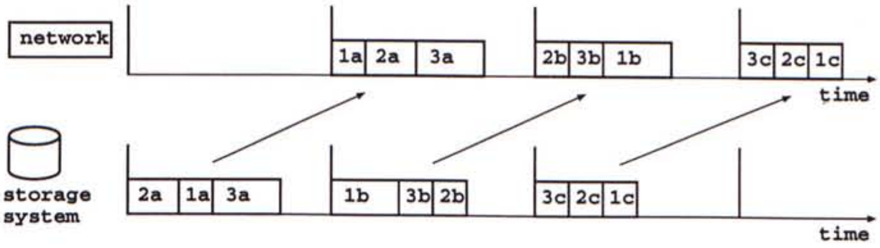

Figure 3.3: Pipelining Effect Between Storage Retrieval and the Network Transmission.

At the end of a disk transfer cycle, the data are available for network transmission. Without loss of generality, we assume the communication network can transmit a fixed amount of data for every period T_0. The data retrieved from the disk storage system will be "packetized" and transmitted across the network. Figure 3.3 illustrates the *pipelining effect* of the data retrieved from the storage system to the transmission network. During the first period, the storage system retrieved three data blocks (e.g., 1a,2a,3a) for these three video streams. These data blocks will be transmitted in the *following* network transmission period. An important point to note is the order of data packet transmission can be different from the order of data retrieval from the storage system. This implies that the storage module and the network transmission module can operate *independently* on selecting which data to retrieve or to transmit (or even to drop, if possible).

Let $\tilde{b_i}$ be the random variable denoting the bandwidth requirement of the i^{th} video request and $b_i = E[\tilde{b_i}]$ be its average bandwidth requirement. One way to guarantee the quality of service for the i^{th} request is to make sure that both the disk storage system *and* the network transmission system can sustain $\tilde{b_i}$, that is, the storage system can retrieve $\tilde{b_i}T$ amount of data per cycle and the network can transmit $\tilde{b_i}T_0$ amount of data per period. To achieve this, one way is to provide *worst-case* resource allocation, that is, let $b_i^* = \max\{\tilde{b_i}\}$, then we need to allocate enough disk I/O bandwidth to retrieve b_i^*T amount of data per cycle and allocate enough transmission bandwidth to transmit $b_i^*T_0$ amount of data per transmission time frame. This type of *pessimistic* admission control usually implies that the resources of the VoD system is under-utilized. The goal of our admission control algorithm is to guarantee the QoS requirements of the individual connection and at the same time, maximize the number of concurrent users that the system can support.

In the following sections, we will present our admission control algorithm. Let us first define some notations which will be used throughout this thesis.

n = number of request currently admitted in the VoD system.
N = number of disk in the storage system.
T_0 = the length of a time frame in the transmission network. This is also the length of a transfer cycle in each disk.
T_i = the length of non-idling time in a transfer cycle of disk $i, i = 1, 2, \ldots, N$.
d_i = the packet dropping rate requirement specified by user i.
p_{ij} = the average probability of a block of video required by viewer j is in disk i.
p = the maximum overflow probability in disk transfer cycle. In order to make higher utilization of disk transfer bandwidth, we permit the disk transfer cycle can overflow with probability less than or equal to this probability.

Note that the average read-size of a video stream should be equal to the average bandwidth requirement of the user who requires this video file times the length of a transfer cycle.

3.3 The Admission Control for the Storage Subsystem

For the n requests currently admitted, the bandwidth requirements are $\mathbf{b} = (b_1, \ldots, b_n)$, the dropping rate requirements are $\mathbf{d} = (d_1, \ldots, d_n)$. The number of required videos in disk i is

$$n_i = \lceil p_{i1} + \cdots + p_{in} \rceil$$

Therefore, for disk $i \in \{1, 2, \ldots, N\}$, we have:

$$T_i = \tau_{seek}^{max}(n_i) + \sum_{j=1}^{n_i} \tau_{rot-j} + \sum_{j=1}^{n_i} \tau_{trf-j}$$

where $\tau_{seek}^{max}(n_i)$ refer to the worst case seek time for service n_i requests, i.e., when these n_i requests are evenly spaced out on the disk surface [6], τ_{rot-j} refers to the rotational latency for video stream j, τ_{trf-j} refers to the disk transfer time for video stream j, $j = 1, \ldots, n$.

Let there be a new request ($(n+1)^{th}$) arriving with bandwidth requirement b_{n+1} and dropping rate requirement d_{n+1}. Assume that the video block requested by the $(n+1)^{th}$ request is in disk i with probability $p_{i(n+1)}, i = 1, \ldots, N$. If it is admitted, T_i becomes:

$$T_i = \tau_{seek}^{max}(n_i') + \sum_{j=1}^{n'} \tau_{rot-j} + \sum_{j=1}^{n'} \tau_{trf-j}$$

where $n_i' = \lceil p_{i1} + \cdots + p_{i(n+1)} \rceil$. Let $F_{D_i}^*(s)$ be the Laplace transform for the random variable T_i and let $F_{rot-j}^*(s)$ and $F_{tfr-j}^*(s)$ be the Laplace transforms for the random variables τ_{rot-j} and τ_{trf-j}, respectively. Since $\tau_{rot-j}, \tau_{trf-j}, j = 1, \ldots, n$ are independent, using the convolution property of Laplace transform, we get:

$$F^*_{D_i}(s) = e^{[-s\tau^{max}_{seek}(n')]} \prod_{j=1}^{n'} F^*_{rot-j}(s) \prod_{j=1}^{n'} F^*_{trf-j}(s)$$

Let $M_i(s)$ be the moment generating function for the random variable T_i. Since $M_i(s)$ is equal to $F^*_{D_i}(-s)$, applying Chernoff's theorem to bound the tail of the random variable T_i, we have the following [8]:

$$Prob[T_i > T_0] \;\leq\; \inf_{\theta \geq 0}\{\frac{M_i(\theta)}{e^{\theta T_0}}\}. \tag{3.1}$$

Using standard numerical solution techniques, we can obtain the optimal θ^* which gives the tightest upper bound. If there exists a T_i such that $\frac{M_i(\theta^*)}{e^{\theta^* T_0}} > p$, the disk transfer cannot satisfy the bandwidth requirement of request $(n+1)$, since it means that at least one disk transfer cycle will violate the overflow probability requirement if $(n+1)^{th}$ request is admitted. On the other hand, if all $\frac{M_i(\theta^*)}{e^{\theta^* T_0}} \leq p$, it means the disk transfer can satisfy the bandwidth requirements of all $n+1$ requests.

Let us consider two special cases of the admission control for storage subsystem under different data placement policies:

Well-balanced Case

The admitted requests are well-balanced, that is, before the $(n+1)^{th}$ request arrives, the required movies are distributed in N disks evenly. So in each disk i, there are at most $\lceil \frac{n}{N} \rceil$ video streams in service.

If we admit the $(n+1)^{th}$ request whose required movie is on disk i with probability $1/N$, $i = 1, \ldots, N$, the cycle time of disk i becomes:

$$T_i = \tau^{max}_{seek}(\lceil \frac{n+1}{N} \rceil) + \sum_{j=1}^{\lceil \frac{n+1}{N} \rceil} \tau_{rot-j} + \sum_{j=1}^{\lceil \frac{n+1}{N} \rceil} \tau_{trf-j}$$

the Laplace transform for T_i becomes

$$F^*_{D_i}(s) \;=\; e^{[-s\tau^{max}_{seek}(\lceil \frac{n+1}{N} \rceil)]} \prod_{j=1}^{\lceil \frac{n+1}{N} \rceil} F^*_{rot-j}(s) \prod_{j=1}^{\lceil \frac{n+1}{N} \rceil} F^*_{trf-j}(s)$$

so using the method mentioned before we can obtain the value of θ^*. If there exists a T_i such that $\frac{M_i(\theta^*)}{e^{\theta^* T_0}} > p$, the disk transfer cannot satisfy the bandwidth requirement of request $(n+1)$. On the other hand, if all $\frac{M_i(\theta^*)}{e^{\theta^* T_0}} \leq p$, it means the disk transfer can satisfy the bandwidth requirements of all $n+1$ requests.

Unbalanced Case:

We consider the case that the requests are extremely unbalanced, that is, before the $(n+1)^{th}$ request arrives, all the movies required are in one disk, say N. The movie required by $(n+1)^{th}$ request is also in Disk N. If we admit request $n+1$, the cycle time of disk N becomes:

$$T_N = \tau_{seek}^{max}(n+1) + \sum_{j=1}^{n+1} \tau_{rot-j} + \sum_{j=1}^{n+1} \tau_{trf-j}$$

and the Laplace transform for T_N becomes:

$$F_{D_N}^*(s) = e^{[-s\tau_{seek}^{max}(n+1)]} \prod_{j=1}^{n+1} F_{rot-j}^*(s) \prod_{j=1}^{n+1} F_{trf-j}^*(s)$$

So using the method we mentioned before we can get θ^* and decide whether the disk transfer can satisfy the bandwidth requirements of all $n+1$ requests or not.

If the bandwidth requirements of users' can be satisfied, then we need check whether the packet dropping rate requirements of the users' can be satisfied or not.

3.4 The Admission Control for Network Subsystem

Before we approach to the admission control algorithm for network sub-system, we need give some preliminaries which we will use in deriving the admission control algorithm for network sub-system.

3.4.1 Preliminaries

Let $E = \{1, \ldots, n\}$ be a finite set; $\mathbf{x} = (x_i)_{i=1}^{n}$ is a n-dimension vector. We first need to define the meaning of polymatroid [17]:

Definition 1 The following polytope

$$\mathcal{P}(f) = \{\mathbf{x} \geq 0 : \sum_{i \in A} x_i \leq f(A), A \subseteq E\} \tag{3.2}$$

is termed a polymatroid if the function $f : 2^E \rightarrow \Re_+$ satisfies the following properties: (i) (normalized) $f(\emptyset) = 0$; (ii) (increasing) if $A \subseteq B \subseteq E$, then $f(A) \leq f(B)$; (iii) (submodular) if $A, B \subseteq E$, then $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$.

Specially, if

$$\mathcal{B}(f) = \{\mathbf{x} \geq 0 : \sum_{i \in A} x_i \leq f(A), A \subset E; \sum_{i \in E} x_i = f(E)\}$$

Then $\mathcal{B}(f)$ is the base of polymatroid $\mathcal{P}(f)$.

The following is an example of polytope(polymatroid):

$$\begin{aligned}
&x_i \geq 0, \qquad i = 1, 2, 3 \\
&x_1 \leq 3 \\
&x_2 \leq 2 \\
&x_3 \leq 4 \\
&x_1 + x_2 \leq 4 \\
&x_1 + x_3 \leq 4 \\
&x_2 + x_3 \leq 4 \\
&x_1 + x_2 + x_3 \leq 4
\end{aligned}$$

The function $f : 2^E \rightarrow \Re_+$ is defined as:

$$\begin{aligned}
f(\emptyset) &= 0 \\
f(\{1\}) &= 3 \\
f(\{2\}) &= 2 \\
f(\{3\}) &= 4 \\
f(\{1,2\}) &= 4 \\
f(\{1,3\}) &= 4 \\
f(\{2,3\}) &= 4 \\
f(\{1,2,3\}) &= 4
\end{aligned}$$

It is easy to show that the function f satisfies the three properties in definition 1, so the above polytope is also a polymatroid and the plane within XYZUV (refer to Figure 3.4) is the base of this polymatroid.

The following definition defines the vertex of the base of polymatroid [4].

Definition 2 Let π denote a permutation of $\{1,2,\ldots,n\}$, $\mathbf{x}^{\pi}$ defined below is a "vertex" of the base of polymatroid defined in Definition 1

$$\begin{aligned}
x^{\pi}_{\pi_1} &= f(\{\pi_1\}) \\
x^{\pi}_{\pi_2} &= f(\{\pi_1,\pi_2\}) - f(\{\pi_1\}) \\
&\vdots \\
x^{\pi}_{\pi_n} &= f(\{\pi_1,\pi_2,\ldots,\pi_n\}) - f(\{\pi_1,\pi_2,\ldots,\pi_{n-1}\})
\end{aligned}$$

Apply definition 2 to the above example, we can get the vertices of $\mathcal{B}(f)$.

when $\pi = \{1,2,3\}$, the vertex is $(3,1,0)$,

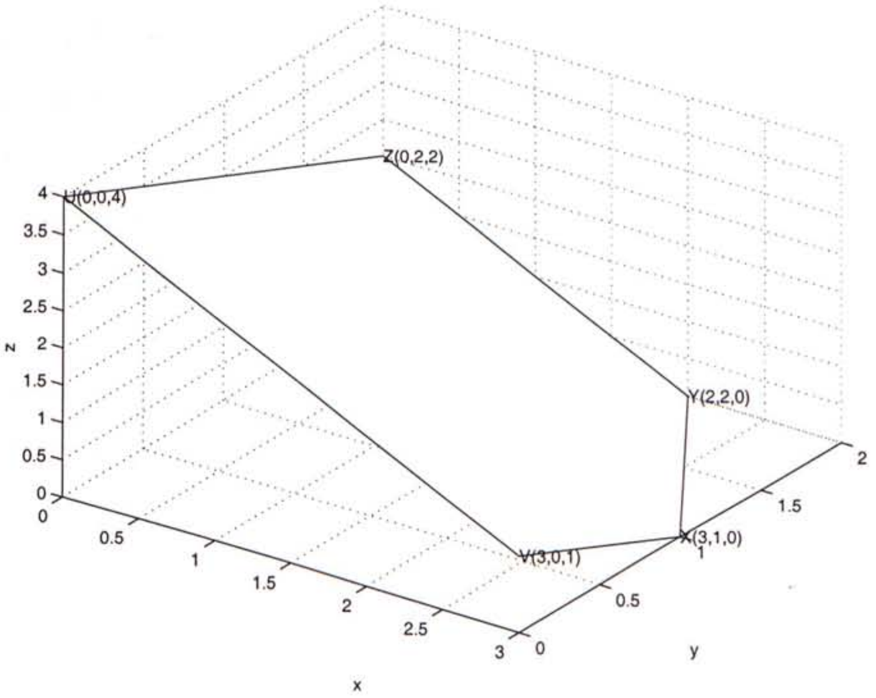

Figure 3.4: The Base of Polymatroid XYZUV

when $\pi = \{1,3,2\}$, the vertex is $(3,0,1)$,

when $\pi = \{2,1,3\}$, the vertex is $(2,2,0)$,

when $\pi = \{2,3,1\}$, the vertex is $(0,2,2)$,

when $\pi = \{3,1,2\}$, the vertex is $(0,0,4)$,

when $\pi = \{3,2,1\}$, the vertex is $(0,0,4)$.

Definition 3 [13] A scheduling control policy is said admissible if it is non-idling and non-anticipative, that is, if no server is allowed to be idle when there are jobs waiting to be served, and the control is only allowed to make use of past history and the current state of the system. Any admissible control cannot affect the arrival processes or the service requirements of the job.

Next, we will give the definition of *strong conservation laws* which was reported in [13].

Let $E = \{1, 2, \ldots, n\}$ denote the set of all job types. For any $A \subseteq E$, let $|A|$ denote the cardinality of A. Let $\mathcal{U}$ denote the set of all admissible policies. Let $\mathbf{x}^u$ denote the performance measure under an admissible policy $u \in \mathcal{U}$. Let $\pi = (\pi_1, \ldots, \pi_n)$ denote a permutation of the integers $\{1, \ldots, n\}$, which represents an admissible priority rule, that is, type π_1 jobs have the highest priority, and type π_n jobs have the lowest priority.

Definition 4 [13] The performance vector $\mathbf{x}$ is said to satisfy *strong conservation laws*, if there exists a set function b (or respectively, f): $2^E \rightarrow \Re_+$,satisfying

$$b(A) = \sum_{\pi_i \in A} x_{\pi_i},\ \forall \pi : \{\pi_1, \ldots, \pi_{|A|}\} = A,\ \forall A \subseteq E; \tag{3.3}$$

or respectively,

$$f(A) = \sum_{\pi_i \in A} x_{\pi_i},\ \forall \pi : \{\pi_1, \ldots, \pi_{|A|}\} = A,\ \forall A \subseteq E; \tag{3.4}$$

(when $A = \emptyset$, by definition, $b(\emptyset) = f(\emptyset) = 0$); such that for all $u \in \mathcal{U}$ the following is satisfied:

$$\sum_{i \in A} x_i^u \geq b(A),\ \forall A \subset E;\ \sum_{i \in E} x_i^u = b(E); \tag{3.5}$$

or respectively,

$$\sum_{i \in A} x_i^u \leq f(A),\ \forall A \subset E;\ \sum_{i \in E} x_i^u = f(E). \tag{3.6}$$

If the performance measure in a particular question is minimized (or maximized) by the admissible priority rules, then the function b (or f) applies in this question [19].

This definition states two requirements that a performance vector must satisfy in order to satisfy strong conservation laws [13]:

1. The summation of all components of the performance vector in question is invariant under any admissible control policy. This requirement is reflected

in following equations

$$\sum_{i\in E} x_i^u = b(E);$$

or

$$\sum_{i\in E} x_i^u = f(E).$$

2. The summation of components of the performance vector in question who represent job types in A is minimized (or maximized) by any absolute priority rule giving the job types in A over the other job types. This requirement is reflected in following equations and inequalities:

$$b(A) = \sum_{\pi_i\in A} x_{\pi_i},\ \forall\pi : \{\pi_1, \dots,\ pi_{|A|}\} = A,\ \forall A \subseteq E;$$

$$\sum_{i\in A} x_i^u \geq b(A),\ \forall A \subset E;$$

or

$$f(A) = \sum_{\pi_i\in A} x_{\pi_i},\ \forall\pi : \{\pi_1, \dots,\ pi_{|A|}\} = A,\ \forall A \subseteq E;$$

$$\sum_{i\in A} x_i^u \leq f(A),\ \forall A \subset E;$$

The following theorem gives the relationship between the strong conservation laws and the base of a polymatroid (We use $\mathcal{B}(b)$ to denote the polytope $\{\mathbf{x} \geq 0 : \sum_{i\in A} x_i \geq b(A), A \subset E; \sum_{i\in E} x_i = b(E)\}$ which is also a base of polymatroid by setting $b(A) := b(E) - f(E - A)$).

Theorem 1 [13] Assume the performance vector $\mathbf{x}$ satisfies the strong conservation laws (3.3) and (3.5) [(3.4) and (3.6)]. Then:

a. the convex polytope $\mathcal{B}(b)$ $[\mathcal{B}(f)]$ is the performance space;

b. $\mathcal{B}(b)$ $[\mathcal{B}(f)]$ is the base of a polymatroid; and

c. the vertices of $\mathcal{B}(b)$ $[\mathcal{B}(f)]$ are the performance vectors of the absolute priority rules.

Since $\mathcal{B}(b)$ $[\mathcal{B}(f)]$ is a convex polytope, any vector in $\mathcal{B}(b)$ $[\mathcal{B}(f)]$ can be expressed as a convex combination of its vertices. This implies that if a performance vector satisfies the strong conservation laws, we can easily derive the space of this performance vector. Also, given a vector, we can easily find out whether there exists an admissible priority rule under which the performance measure can achieve this vector or not.

3.4.2 The Admission Control Algorithm for Network Subsystem

In a transmission time frame, assuming the packets which need to be transmitted in current time frame arrive at the network at the beginning of current time frame. We employ admissible policy defined in last subsection since obviously, allowing the server idling when there are packets need to be transmitted in current time frame will not benefit the system. Also, in practice, it is usually difficult to obtain the future information of the system, so an admissible policy is desirable. Here we regard packets which should be transmitted in current time frame but can not be transmitted as violating the deadline and must be dropped. Since the probability of overflow in a disk transfer cycle is very small and we can ignore the bits dropped in disk transfer cycle, we can assume that the distribution of the size of video stream which need to be transmitted in a transmission time frame is the same as the disk read-size in the corresponding disk transfer cycle.

Assume the packet size is same for all the video streams(this is the case in ATM networks where each packet(cell) has the fixed length), it is easily seen that:

1. The overall dropping rate, over all video streams in E is invariant under

any admissible policy;

2. The dropping rate over any given subset $A, A \subset E$ is minimized by offering absolute priority to video streams in the subset A over the video streams in $(E - A)$.

So we know that the dropping rate vector $\mathbf{d} = (d_1, \ldots, d_{n+1})$ satisfies the strong conservation laws.

From the preliminaries, we know the space of dropping rate $\mathbf{d}$ is a base of polymatroid with each vertex corresponding to a specific absolute priority scheduling policy. Since this base of polymatroid is a convex polytope, any point of this base of polymatroid can be expressed as a convex combination of its vertices. So if the required dropping rate vector is in this polytope, we know we can find a convex combination of absolute priority rules to achieve it.

Let $low(A)$ denote the lower bound of dropping rate over given subset A, $A \subset E$. If the dropping rate vector $\mathbf{d} = (d_1, \ldots, d_{n+1})$ satisfies[3]:

$$\sum_{i \in A} d_i \geq low(A),\ A \subset E$$

and

$$\sum_{i \in E} d_i > low(E)$$

we can always find a point in the space of $\mathbf{d}$ which is better than given $\mathbf{d}$ and can be realized by a convex combination of absolute priority rules.

Now we consider a simple example to illustrate the above idea:

Example 1 Suppose currently there are 2 requests admitted in the VoD system. The arrival rate of the first video stream is 5 packets/time frame, the arrival rate of the second video stream is 7 packets/time frame with

---

[3] In here, $E = \{1, \ldots, n+1\}$

probability 0.5 and 3 packets/time frame with probability 0.5. The maximum number of packets can be transmitted in a time frame is 11. So we have $low(\{1\}) = low(\{2\}) = 0, low(\{1,2\}) = 0.5$. Applying definition 2 by setting $low(A) := low(E) - f(E - A)$, we can get the admissible region for dropping rate $\mathbf{d} = (d_1, d_2)$ which is illustrate in Figure 3.5. So if the dropping rate requirement is $\mathbf{d} = (0.5, 1)$, it is easily seen that $Z(0.5, 1)$ is in the admissible region but not in the space of $\mathbf{d}$. So we can achieve a better dropping rate vector $(0.5, 0)$ by giving higher priority to video stream 2.

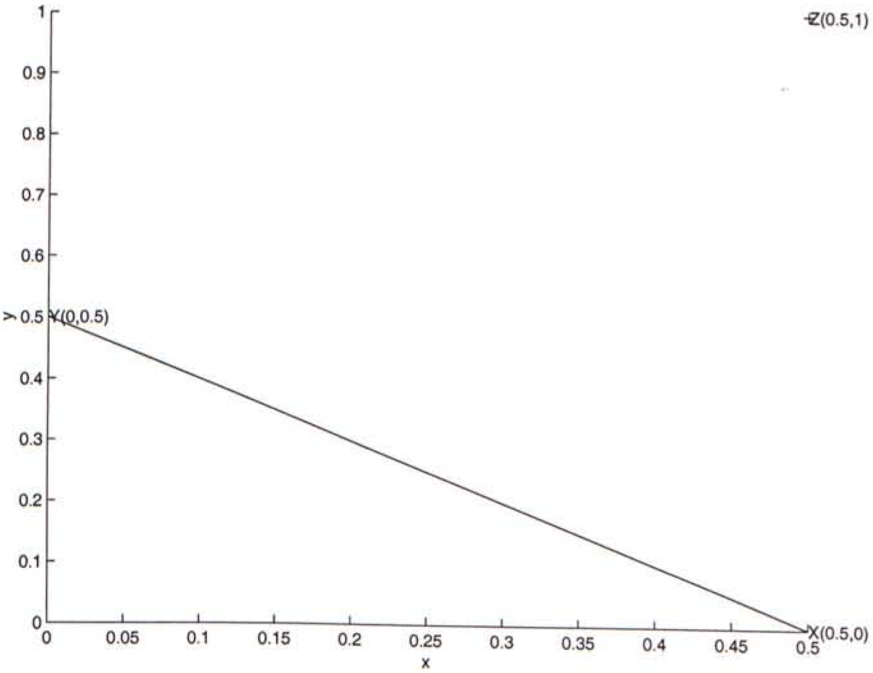

Figure 3.5: The Admissible Region of the Dropping Rate Vector (d_1, d_2) in Example 1.

Now consider the third requested video stream is arriving with rate 5 packets/time frame with probability 0.5 and 3 packets/time frame with probability 0.5. So if the third request is admitted, we can get:
$low(\{1\}) = low(\{2\}) = low(\{3\}) = 0, low(\{1,2\}) = 0.5,$
$low(\{1,3\}) = 0, low(\{2,3\}) = 0.25, low(\{1,2,3\}) = 3.$

The admissible region of the dropping rate vector $\mathbf{d} = (d_1, d_2, d_3)$ includes all the points (x_1, x_2, x_3) that satisfy:

$$\begin{aligned}
&x_i \geq 0, \qquad i = 1, 2, 3\\
&x_1 + x_2 \geq 0.5\\
&x_1 + x_3 \geq 0\\
&x_2 + x_3 \geq 0.25\\
&x_1 + x_2 + x_3 \geq 3
\end{aligned}$$

The space of **d** is illustrated in Figure 3.6. The vertices of the space of **d** are: $X(2.75, 0.25, 0), Y(2.75, 0, 0.25), Z(0.5, 0, 2.5), U(0, 0.5, 2.5), V(0, 3, 0)$. The corresponding priority rules to achieve these vertices are:$(3, 2, 1), (2, 3, 1)$, $(2, 1, 3), (1, 2, 3), (3, 1, 2)$.

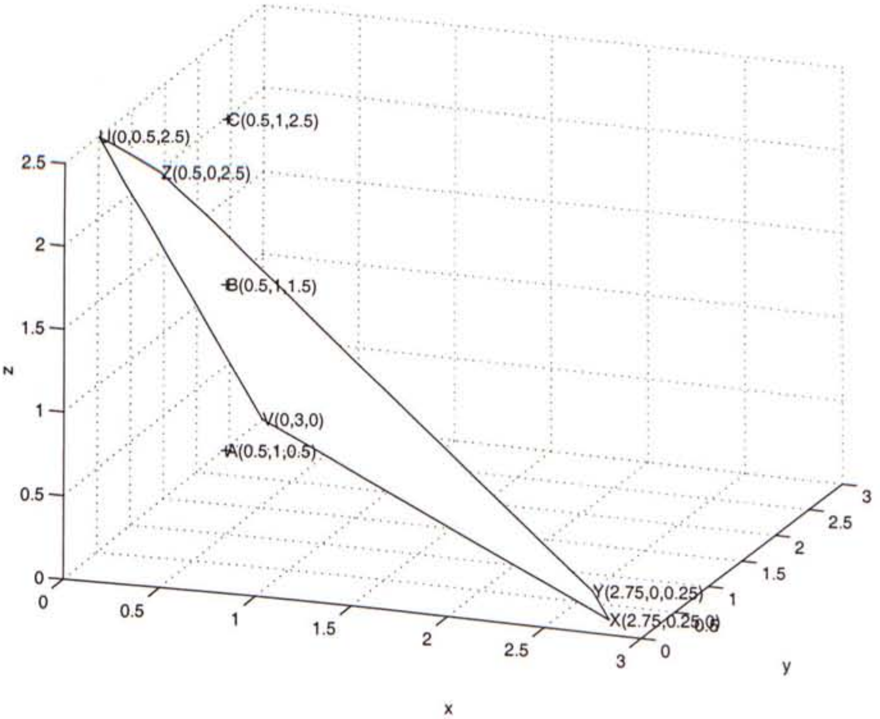

Figure 3.6: The Space of the Dropping Rate vector (d_1, d_2, d_3) in Example 1.

Let us consider the following 3 scenarios:

1. If the third request has the dropping rate requirement $d_3 = 0.5$, the dropping rate requirement is $A(0.5, 1, 0.5)$. Because the new total dropping rate requirement is $d_1 + d_2 + d_3 = 0.5 + 1 + 0.5 = 2$, which is smaller than $low(\{1, 2, 3\})(= 3)$, we cannot find an admissible policy to achieve it. The third request will be rejected.

2. If the third request has the dropping rate requirement $d_3 = 1.5$, the dropping rate requirement is $B(0.5, 1, 1, 5)$. Since $(0.5, 1, 1.5)$ is an internal point of the convex polytope in Figure 3.6 , we know we can find a convex combination of absolute priority rules to achieve it. Specifically, by solving the linear equation systems:

$$a_1 \begin{pmatrix} 0.5 \\ 0 \\ 2.5 \end{pmatrix} + a_2 \begin{pmatrix} 2.75 \\ 0 \\ 0.25 \end{pmatrix} + a_3 \begin{pmatrix} 2.75 \\ 0.25 \\ 0 \end{pmatrix} + \\ + a_4 \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix} + a_5 \begin{pmatrix} 0 \\ 0.5 \\ 2.5 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1 \\ 1.5 \end{pmatrix}$$

   where $\sum_{i=1}^{5} a_i = 1$ and $a_i \geq 0,\ i = 1, 2, \ldots, 5$. We can get a vector $(a_1, a_2, a_3, a_4, a_5) = (l, k, \frac{2}{11} - \frac{2}{11}l - k, \frac{12}{55} + \frac{2}{11}l + \frac{1}{10}k, \frac{3}{5} - l - \frac{1}{10}k)$, where $0 \leq l, k \leq 1$ and $l + k \leq 1$. The admissible policies which can achieve the dropping rate $(0.5, 1, 1.5)$ is

$$l(2,1,3) + k(2,3,1) + (\frac{2}{11} - \frac{2}{11}l - k)(3,2,1) + \\ + (\frac{12}{55} + \frac{2}{11}l + \frac{1}{10}k)(3,1,2) + (\frac{3}{5} - l - \frac{1}{10}k)(1,2,3)$$

3. If the third request has the dropping rate requirement $d_3 = 2.5$, the dropping rate requirement is $C(0.5, 1, 2.5)$. We can prove that

$$\sum_{i \in A} d_i \geq low(A), A \subset E \quad \text{and} \quad \sum_{i \in E} d_i > low(E)$$

So we know we can find an admissible policy which can achieve a better dropping rate vector than $\mathbf{d} = (0.5, 1, 2.5)$. Specifically, we can use the admissible policy derived in case 2 to achieve a better dropping rate vector $(0.5, 1, 1.5)$. Therefore, the third request can be admitted.

Chapter 4

Experiment

In this chapter, we present some experimental results obtained by using our proposed admission control algorithm. We also compare these results with the results obtained by using average bandwidth allocation strategy. Since using peak bandwidth allocation (or worst case resource allocation) will result in worst bandwidth utilization than that of using average bandwidth allocation, we will not include the comparison between our method and peak bandwidth allocation strategy.

In our first experiment, there are two disks in the VoD storage system. The related characteristics of each disk are listed in Table 4.1.

In Table 4.1, d is the seek distance for a requested video block in a disk transfer cycle. We use the worst case seek distance as the value of d, which is the seek distance for a requested video block when all the requested video blocks

Number of cylinders	5288
Transfer rate	80 Mbps
Maximum rotational latency	8.33 milliseconds
Seek time function (secs)	$seek(d) = \begin{cases} 0.6 * 10^{-3} + 0.3 * 10^{-3} * \sqrt{d} & \text{if } d < 400 \\ 5.75 * 10^{-3} + 0.002 * 10^{-3} * d & \text{if } d \geq 400 \end{cases}$

Table 4.1: The Parameters of the Disk Used in the Experiment

in a disk are evenly spaced out on the surface of this disk. So

$$d = \lceil \frac{\text{Number of cylinders in the disk}}{\text{Number of requested video blocks in the disk}} \rceil$$

and $seek(d)$ is the seek time for finding a video block in a disk. Also, we assume the rotational latency for the requested video block is uniformly distributed in the range $[0, 8.33]$ millisecond.

The compressed video are stored in the disks using RAID which we talked about in Chapter two. The size of one data block is exponentially distributed with mean 1.5 Mb for each MPEG-1 video block and 5 Mb for each MPEG-2 video block.

The VoD system can provide service in three classes. Class $\mathcal{A}$ service is provided to users whose requested videos are compressed by MPEG-1 and whose average bandwidth requirements are 1.5 Mbps and dropping rate requirements are 2% (that is, the average number of bits dropped in a transmission time frame should be less than $1.5 * 2\% = 0.03$ Mb). Class $\mathcal{B}$ service is provided to users whose requested videos are compressed by MPEG-1 and whose average bandwidth requirements are 1.5 Mbps and dropping rate requirements are 6%. Class $\mathcal{C}$ service is provided to users whose requested videos are compressed by MPEG-2 and whose average bandwidth requirements are 5 Mbps and dropping rate requirements are 10% [18]. We call a user's request "class i request" if the user will get class i service when his request is admitted, $i \in E = \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$. The length of a transfer cycle in the disk is 1 second. During each transfer cycle, the read-size of video is exponentially distributed with mean 1.5 Mb for each MPEG-1 video and 5 Mb for each MPEG-2 video. The given overflow probability of disk transfer cycle is $1.0 * 10^{-4}$. The transmission network we employed is ATM network with bandwidth 155 Mbps. The length of a transmission time frame is 1 second.

From the previous chapters, we know that a new request (or new requests)can be admitted if and only if the system has enough resource to satisfy the QoS requirements of the new request(s) and at the same time, the QoS requirements of other existing viewers will not be violated.

We assume that at the beginning of the experiment, there are 51 video streams in the VoD system. Each class has 17 video streams (one can easily verify that the VoD system can support these video streams, both in disk sub-system and the network sub-system). Now there are one class $\mathcal{A}$ request, two class $\mathcal{B}$ requests and one class $\mathcal{C}$ request arrive at the same time. To see whether these new requests can be admitted or not, first we need check whether the bandwidth requirements of the users' can be satisfied. Assuming the admitted requests are well-balanced, if the new requests are admitted in disk sub-system, applying the formula in Table 4.1, the worst case seek time for a video is 0.0047 secs, then applying Equation (3.1), we got

$$P[T_i \geq 1] \leq 7.7539 * 10^{-5} \quad i = 1, 2$$

so the probability of overflow of the disk transfer cycle is less that given $p = 1.0 * 10^{-4}$. The bandwidth requirements of all users' can be satisfied.

Next, we need check whether the dropping rate requirements of all users' can be satisfied. We treat the video streams belonging to the same service class as one "big" video stream and compute the lower bound of dropping rate $low(A)$, where $A \subseteq E, E = \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ represents the set of three "big" video streams. The results are listed in Table 4.2, also listed are dropping rate requirements $d(A), A \subseteq E$. By comparing the dropping rate requirements $d(A)$ with $low(A)$, we got:

$$d(A) \geq low(A) \quad \text{where } A \subseteq E,\ E = \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$$

So these new requests can be admitted. The admissible region includes all the

Item	Value(Mb)	Item	Value(Mb)
$d\{\mathcal{A}\}$	0.54	$low\{\mathcal{A}\}$	1.2397e-25
$d\{\mathcal{B}\}$	1.71	$low\{\mathcal{B}\}$	7.2971e-25
$d\{\mathcal{C}\}$	9	$low\{\mathcal{C}\}$	0.0442
$d\{\mathcal{AB}\}$	2.25	$low\{\mathcal{AB}\}$	3.7213e-14
$d\{\mathcal{AC}\}$	9.54	$low\{\mathcal{AC}\}$	0.6438
$d\{\mathcal{BC}\}$	10.71	$low\{\mathcal{BC}\}$	0.7337
$d\{\mathcal{ABC}\}$	10.764	$low\{\mathcal{ABC}\}$	5.4121

Table 4.2: $d(A)$ and $low(A)$ in Experiment 1

points whose coordinates (x_1, x_2, x_3) satisfy:

$$
\begin{aligned}
&x_i \geq 0, \qquad i = 1, 2, 3 \\
&x_1 \geq low(\mathcal{A}) \\
&x_2 \geq low(\mathcal{B}) \\
&x_3 \geq low(\mathcal{C}) \\
&x_1 + x_2 \geq low(\mathcal{AB}) \\
&x_1 + x_3 \geq low(\mathcal{AC}) \\
&x_2 + x_3 \geq low(\mathcal{BC}) \\
&x_1 + x_2 + x_3 \geq low(\mathcal{ABC})
\end{aligned}
$$

The space of **d** is illustrate in Figure 4.1 where HIJKO is a base of polymatroid, $H = (4.6783, 0.6895, 0.0442), I = (4.6783, 0, 0.7337), J = (0.5996, 4.7682, 0.0442)$, $K = (0, 4.7682, 0.6438), O = (0, 0, 5.4121)$. The corresponding admissible policy to achieve these vertices are: (3,2,1), (2,3,1),(3,1,2),(1,3,2),(2,1,3).

From the previous chapter, we know we can find the admissible scheduling policy to achieve each point in the base of polymatroid HIJKO. For each of other points in admissible region, we can find an admissible scheduling policy to achieve a "better" point than the point itself.

Because $d(E) = 10.764 > 5.4121 = low(E)$, we know we can find a point which is in the base of polymatroid HIJKO and which is better than the given

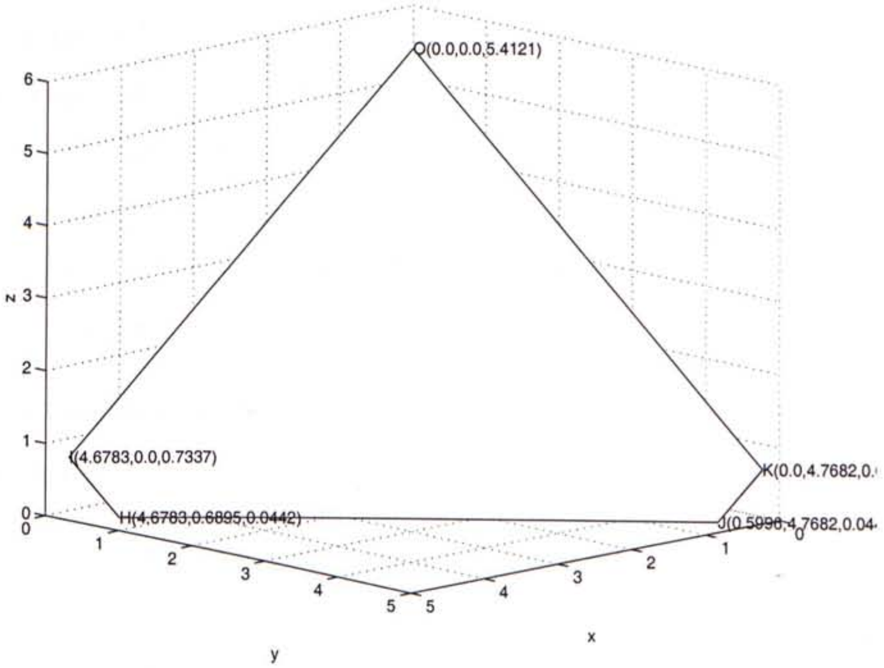

Figure 4.1: The Base of Polymatroid HIJKO which is the space of **d**

dropping rate vector. Here, the given dropping rate vector is $(0.54, 1.71, 9)$ (see Table 4.2), by deducting 1 from d_2, deducting 4.8379 from d_3, we got the "better" point $(0.54, 0.71, 4.1621)$, where $0.54 + 0.71 + 4.1621 = 5.4121 = low(E)$. After solve the following linear equation systems,

$$a_1 \begin{pmatrix} 4.6783 \\ 0.6895 \\ 0.0442 \end{pmatrix} + a_2 \begin{pmatrix} 4.6783 \\ 0.0 \\ 0.7337 \end{pmatrix} + a_3 \begin{pmatrix} 0.5996 \\ 4.7682 \\ 0.0442 \end{pmatrix} +$$

$$+ a_4 \begin{pmatrix} 0.0 \\ 4.7682 \\ 0.6438 \end{pmatrix} + a_5 \begin{pmatrix} 0.0 \\ 0.0 \\ 5.4121 \end{pmatrix} = \begin{pmatrix} 0.54 \\ 0.71 \\ 4.1621 \end{pmatrix}$$

$$\sum_{i=1}^{5} a_i = 1$$

$$a_i \geq 0, \ i = 1, 2, \ldots, 5$$

we got $(a_1, a_2, a_3, a_4, a_5) = (0, 0.0963, 0.1489, 0, 0.7548)$, that is, the admissible policy to achieve $(0.54, 0.71, 4.1621)$ is the policy that at the beginning of a time frame, with probability 0.0963, the scheduling policy is (2,3,1); with probability 0.1489, the scheduling policy is (3,1,2); with probability 0.7548, the scheduling policy is (2,1,3). Within each class, the scheduling policy is allocating the bandwidth proportional to the number of bits each video stream has at the beginning of the time frame.

We continue doing the experiment, and got the result that at least the considered VoD system can support 20 video streams for class $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$, respectively.

Now we compare the results obtained by using our proposed algorithm with the results obtained by average bandwidth allocation strategy. Here "average bandwidth allocation strategy" refers to the strategy that allocates transmission bandwidth to the video streams in network sub-system according to the average bandwidth requirement of the users'. This strategy also considers the VBR nature of the video streams, so it can achieve higher bandwidth utilization than peak bandwidth allocation strategy. In our example, employing average bandwidth allocation strategy means allocating 1.5 Mbps bandwidth to each video streams in class $\mathcal{A}$ and class $\mathcal{B}$ and 5 Mbps bandwidth to each video streams in class $\mathcal{C}$. If there are 20 video streams for class $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$, respectively, the overall bandwidth requirement is 160 Mbps which exceeds the bandwidth of the transmission network which is 155 Mbps. So by employing average bandwidth allocation strategy, we cannot admitted 20 video streams for class $\mathcal{A}, \mathcal{B}, \mathcal{C}$, respectively. And we cannot determine whether the dropping rate requirement of each user's can be satisfied or not. This is in contrast with our proposed algorithm which can accommodate 20 video streams for class $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$, respectively, and at the same time, we can find the admissible scheduling policy to achieve the dropping rate requirements easily.

In our second experiment, we use the same type of disks in experiment 1, but the number of disks is 9.

The videos are compressed using MPEG-2 and stored in the disks using RAID. There are three kinds of videos. The bandwidth requirements of the first, second and third kind of videos are exponentially distributed with mean 5, 10, 15 Mbps, respectively. This is corresponding to the situation that more and more videos are compressed using MPEG-2 and the bandwidth requirement of MPEG-2 videos range from 4 Mbps to 15 Mbps.

The VoD system can provide service in three classes. Class $\mathcal{A}$ service is provided to users whose average bandwidth requirements are 5 Mbps and dropping rate requirements are 10% [18]. Class $\mathcal{B}$ service is provided to users whose average bandwidth requirements are 10 Mbps and dropping rate requirements are 10%. Class $\mathcal{C}$ service is provided to users whose average bandwidth requirements are 15 Mbps and dropping rate requirements are 10%. The length of a transfer cycle in the disk is 1 second. During each transfer cycle, the read-size of videos is exponentially distributed with mean 5 Mb, 10 Mb and 15 Mb for class $\mathcal{A}, \mathcal{B}, \mathcal{C}$ service, respectively. The given overflow probability of disk transfer cycle is $1.0 * 10^{-4}$. The transmission network we employed now is ATM network with bandwidth 622 Mbps. The length of a transmission time frame is 1 second.

Assuming the admitted requests are well-balanced, using our admission control algorithm, we know that the VoD system can support 60 video streams: 10 class $\mathcal{A}$ streams, 20 class $\mathcal{B}$ streams, 30 class $\mathcal{C}$ streams.

Now we compare the results obtained by using our proposed algorithm with the results obtained by average bandwidth allocation strategy. Using average bandwidth allocation strategy means allocating 5 Mbps, 10 Mbps and 15 Mbps bandwidth to each video streams in class $\mathcal{A}, \mathcal{B}, \mathcal{C}$, respectively. If there are 10, 20, 30 video streams for class $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$, respectively, the overall bandwidth requirement is 700 Mbps which exceeds the bandwidth of the transmission network which is 622 Mbps. So the ATM network can not support 10 class $\mathcal{A}$, 20 class $\mathcal{B}$ and 30 class $\mathcal{C}$ video streams. Also, we cannot determine whether the dropping rate requirement of each user's can be satisfied or not. This is in contrast with our proposed algorithm which can accommodate 10,20,30 video streams for class $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$, respectively, and at the same time, we can find the scheduling policy to achieve the dropping rate requirements easily.

Note that employing our admission control algorithm, we can check several simultaneously arriving requests at the same time. When the video streams become bursty, the benefit of using our admission control algorithm becomes larger (that is, the utilization of system resource will be significantly higher than that by using the average bandwidth allocation strategy). Also note that, the time complexity of our algorithm is a function of number of service class, not the number of total requests. So the time complexity of our algorithm will be fixed for a specific VoD system with fixed number of service class.

Chapter 5

Conclusion and Future Work

In this chapter, we give the conclusion that our proposed admission control algorithm can guarantee the QoS requirements of individual connections and at the same time, achieve high bandwidth utilization. We also discuss some future work.

5.1 Conclusion

In this thesis, we considered a theoretical framework of performing admission control in a VoD system. Previous work on the admission control in VoD is usually performed on a aggregated traffic basis. Our admission control algorithm can guarantee the QoS requirements of individual connections and at the same time, achieve high bandwidth utilization. In the storage sub-system, some form of cycle-based scheduling algorithm is used and we determine the conditions in which the storage sub-system can satisfy the bandwidth requirements of the users'. For the network sub-system, we derive the admissible region of the packet dropping rate vector and the admissible scheduling policy to achieve the dropping rate vector within the admissible region.

Our admission control algorithm for network sub-system is an application of strong conservation laws in VoD system. Strong conservation laws are one kind of fundamental laws for a wide range of stochastic systems [19]. If a performance vector satisfies the strong conservation laws, its space is a base of polymatroid. Each vertex in this base of polymatroid can be achieved by an absolute priority rule. This gives us a clear concept of the admissible region for the dropping rate vector in network sub-system of VoD system, which will be very useful if we want to find an optimal scheduling policy to achieve a vector in it.

Experiments show that our proposed algorithm can achieve high bandwidth utilization, making VoD service cost-effectively.

5.2 Future Work

In this thesis, we assume the read-size of the video streams in disk is a kind of random variable whose Laplace transform is known or can be got. This model belongs to the stochastic source model. Instead of stochastic source models, video sources can be specified by stochastic bound or deterministic time-invariant traffic envelope. How to derive the efficient admission control algorithm for VoD system under these two video source models is an interesting problem.

With the popular of Internet, the combination of the Internet and VoD may provide many new applications on entertainment, business and education in the future [7]. An important issue in providing VoD service over Internet is how to adapt the admission control algorithm to the newly proposed diffserv model so that the individual viewer's QoS requirements can be satisfied.

Bibliography

[1] S. Berson, L. Golubchik, and R. R. Muntz. Fault tolerant design of multimedia servers. *Proc. of the ACM SIGMOD Conf. on Management of Data*, pages 364–375, May 1995.

[2] M.J. Capone and I. Stavrakakis. Achievable qos and scheduling policies for integrated services wireless networks. *Performance Evaluation*, 27 & 28:347–365, 1996.

[3] M. Chen, D. Kandlur, and P. Yu. Optimization of the grouped sweeping scheduling with heterogeneous multimedia streams. *ACM Multimedia*, pages 235–242, 1993.

[4] J. Edmonds. Submodular functions, matroids and certain polyhedra. *Prof. Int. Conf. on Combinatorics*, 1970.

[5] D.J. Gemmell, H.M. Vin, D.D. Kandlur, P.V. Rangan, and L.A. Rowe. Multimedia storage servers: A tutorial. *IEEE Computer*, 28:40–49, May 1995.

[6] S. Ghandeharizadeh and R.R. Muntz. Design and implementation of scalable continuous media servers. *Special issue of Parallel Computing Journal on Parallel Data Servers and Applications*, 1998.

[7] D. Ghose and H.J. Kim. Scheduling video streams in video-on-demand systems: a survey. *Multimedia Tools and Applications*, 11:167–195, 2000.

[8] L. Golubchik, John C.S. Lui, Edmundo de Souza E. Silva, and R.R. Gail. Evaluation of tradeoffs in resource management techniques for multimedia storage servers. *IEEE International Conference on Multimedia System*, June 1999.

[9] X. Jiang and P. Mohapatra. Efficient admission control algorithms for multimedia servers. *Multimedia Systems*, 7:294–304, 1999.

[10] M. Krunz. Bandwidth allocation strategies for transporting variable-bit-rate video traffic. *IEEE Communications Magazine*, pages 40–46, January 1999.

[11] M. Krunz and S.K. Tripathi. Impact of video scheduling on bandwidth allocation for multiplexed mpeg streams. *Multimedia Systems*, 5:347–357, 1997.

[12] K. Lee and H.Y. Yeom. An effective admission control mechanism for variable-bit-rate video streams. *Multimedia Systems*, 7:305–311, 1999.

[13] J.G. Shanthikumar and D.D. Yao. Multiclass queueing systems: Polymatroidal structure and optimal scheduling control. *Operations Research*, 40, Suppl. No. 2:S293–299, 1992.

[14] A.S. Tanenbaum. *Computer Networks.* Prentice Hall, Upper Saddle River, N.J., 1996.

[15] T.J. To and B. Hamidzadeh. Dynamic real-time scheduling strategies for interactive continuous media servers. *Multimedia Systems*, 7:91–106, 1999.

[16] F. A. Tobagi, J. Pang, R. Baird, and M. Gang. Streaming raid - a disk array management system for video files. *ACM Multimedia Conference*, pages 393–399, 1993.

[17] D. Welsh. *Matroid Theory.* Academic Press, London, 1976.

[18] D. Wijesekera, J. Srivastava, A. Nerode, and M. Foresti. Experimental evaluation of loss perception in continuous media. *Multimedia Systems*, 7:486–499, 1999.

[19] D.D. Yao and L. Zhang. Stochastic scheduling via polymatroid optimization. *Lectures in Applied Mathematics*, 33:333–364, 1997.

[20] P. S. Yu, M.-S. Chen, and D. D. Kandlur. Design and analysis of a grouped sweeping scheme for multimedia storage management. *Third International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 44–55, 1992.

[21] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83:1374–1396, 1995.

[22] G.K. Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology.* Addison-Wesley, Cambridge, MA, USA, 1949.