SUPPORT FOR SERVICE SCALABILITY IN VIDEO-ON-DEMAND END-SYSTEMS

by

Emmanuel Lazare Abram-Profeta

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
1998

Doctoral Committee:
Professor Kang G. Shin, Chair
Associate Professor Atul Prakash
Assistant Professor Nandit R. Soparkar
Professor Toby J. Teorey
Assistant Professor Kimberly M. Wasserman

UMI Number: 9840492

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

To my sweet wife Lisa.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Professor Kang G. Shin, for his guidance, constant encouragement, and generous support, which benefited not only my Ph.D. process but also my career to come. I would also like to express my appreciation to other committee members. Professors Atul Prakash. Nandit Soparkar. Toby Teorey. and Kimberly Wasserman for their time and constructive comments.

A note of thanks to fellow students of the Real-Time Computing Laboratory. especially Anees Shaikh and Atri Indiresan, for their help and friendship. I also want to thank Beverly J. Monaghan, Dawn Freysinger. and Karen Liska who always went beyond the call of duty to be helpful.

I gratefully acknowledge the National Science Foundation and the Office of Naval Research for providing support during the course of my graduate program.

Thank you to Bruce Meier. Bernard Bunner, Hugues Blangy. Wei Chuan Goh. and Thomas Fontaine for their constant friendship and encouragement. I wish to express my special thanks to Susan and Ron Johnson. who have always considered me as their own son. Finally, my deepest gratitude goes to my wife Lisa, to whom this work is dedicated. Her unconditional love, support, and understanding throughout all these years made this dissertation a reality.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

LIST OF APPENDICES

APPENDIX

ABSTRACT

Large-scale deployment of video-on-demand (VoD) systems depends on the quality-of-service (QoS) in customers' interactions. These interactions typically involve the *VoD end-systems*, that is, the local VoD server and customers' premise equipment (CPE), which communicate via an access network. At this level, factors such as storage reconfigurations and time variations in the request arrival pattern usually cause customers' QoS degradation. This dissertation determines how to achieve *service scalability*, which is defined as the ability to accommodate possibly significant workload fluctuations and overloading situations without affecting customers' QoS in interactions.

First, VoD server storage should be organized to (1) make a large collection of movie titles and concurrent channels available at the lowest possible cost, and (2) offer the lowest admission latency, or minimum waiting time for service. We show that the coarse-grained striping scheme in disk arrays is a potential candidate for its cost-effective storage usage and low admission latency. In addition, we identify the feasibility of clustered disk-array-based storage organizations to reduce service disruptions during content updates. For a given storage organization, scalability can be further improved by *batching* customers' requests and using multicast communication. For instance, in "Near-VoD" (NVoD), videos are sourced at equally-spaced intervals, thus allowing limited VCR functionality and guaranteeing a specified maximum admission latency. In order to fully exploit these features, an analytical study of NVoD is presented, which compares customers' and service provider's conflicting objectives.

If, unlike in NVoD, batching is done without any control in the channel allocation process, extended periods of demand surge may lead to congestion cycles, causing a scalability problem. We thus introduce a methodology to measure scalability and identify scalable alternatives to NVoD. These multicast VoD systems are applicable in a wide range of clustered storage organizations, and may offer a better tradeoff between customers' QoS and server's throughput. However, in these multicast VoD systems, continuity in VCR actions can only be provided intermittently. To alleviate this limitation, we propose and evaluate mechanisms which greatly improve customers' VCR functionality with minimal degradation of service scalability.

CHAPTER 1

INTRODUCTION

1.1 Background

Advances in video compression, video servers, and networking have stimulated the frenzy of alliances between cable, movie studios, and telephone companies, service providers and equipment manufacturers. Seen today as one of the most lucrative new markets emerging from such corporate activities, large-scale digital video service has the potential to replace neighborhood videotape rental stores, and provide convenient and unrestricted access to a large collection of movie titles. Considering the costs in networking, storage, and customers' premise equipments (CPEs), a mass market is needed for the technology to be viable [90]. To support such deployment (e.g., 10,000 – 100,000 customers viewing 2,000 – 20,000 movies), it is usually recognized [15, 26, 44, 63, 65, 74] that a hierarchical video-on-demand (VoD) server architecture is preferable for scalability and economical viability.

In a generic metropolitan architecture depicted in Figure 1.1, service providers manage their own set of video materials, disseminated from one or more archives to a set of front-end local video servers via high-speed regional and backbone networks which are likely to be an ATM variety. Movie archives serve as a repository for a large collection of movie titles, compressed and stored in video databases on secondary and possibly tertiary storages. These archives are used by service providers to periodically update local video servers, during off-peak hours [65]. Caching programs in local VoD servers avoids the high cost of network infrastructure and core network bandwidth demands that would otherwise incur when a large number of customers are served from a single point [15]. Service providers are also responsible for (i) balancing loads among local VoD servers and (ii) video migration between different levels of the hierarchy so as to maximize the probability of honoring a customer's request locally. One can take advantage of locality and time variations in request

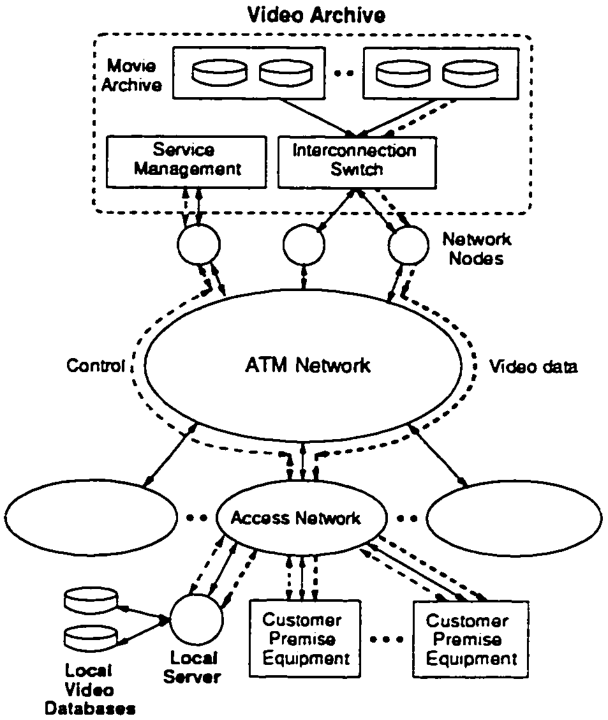


Figure 1.1: A metropolitan video-on-demand system architecture.

arrival rates in conjunction with high-speed networking to improve customers' service by storing frequently-accessed videos on a faster, more expensive storage while keeping less frequently-requested movies on slower, less expensive archives [15, 44, 63, 74]. In the rare case of requests for movie titles not available locally, nearby local VoD servers can be queried. Alternatively, data can be transferred at high speeds from the closest archive and cached locally for playout.

This dissertation deals with the interactive aspect of VoD. For the most popular movies, customers' level of interactivity is affected by the architectural decisions for the choice of a storage medium and resource allocation in local VoD servers. Consequently, the scope of our research will be at this level. Literally speaking, the term video-*on-demand* suggests that ideally, customers' interactions with the service provider should be satisfied *at will*. In the traditional pay-per-view (PPV) service provided by broadcast cable companies, there is no interaction since subscribers "tune in" to pre-scheduled programs, on which they have no control. The most basic improvement over this type of service is to provide users with the VCR playback capability of rental programs. Such service is called "deterministic" VoD (D-VoD) if no customer interaction is allowed after the start of a program, because movies are of constant length, and hence, customers receive deterministic service. A service is known as "true" VoD (T-VoD) if it provides VCR functionality [65]. In between these two extremes, variations on the basic pay-per-view scheme have been proposed in the literature [52, 65], depending on the degree to which VoD services make use of user interactivity.

At the local level, a basic VoD interaction scenario consists of a movie database perusal or query, a request for a feature presentation, and movie viewing. This process typically involves the *VoD end-systems*, that is, the local VoD server, and CPEs, which communicate via an access network. The latter requires a technology that will allow several Mbps to be transmitted to individual homes, since typically, VHS-quality MPEG-1 [61, 82, 83, 84] and HDTV-quality MPEG-2 channels require from 1.5 to 16 Mbps. Possible alternatives include asymmetric digital subscriber line (ADSL), fiber-to-the-home (FTTH), fiber-to-the-curb (FTTC), and hybrid-fiber-coax (HFC). Let's briefly review each of these technologies. ADSL has the advantage that modems can attach to a standard telephone line, and thus be used by businesses that do not have cable TV service to offer interactive services. The required modem technology at each end of the connection is, however, currently very expensive. In addition, even though the maximum input data rate has been increased from 1.5 Mbps in 1993 to 50 Mbps in 1995 [40], a maximum output data rate 640 Kbps makes the system unsuitable for jobs where large amounts of data must flow in both ways. FTTH and FTTC require running high-speed fiber-optic all the way into individual homes or to curbside vaults that serve a dozen or so households. In the latter case, the optical signal that passed through the fiber-optics is converted at vaults into an electrical signal and sent down coaxial cables into customers' homes. These two options are prohibitively expensive, especially considering the billions of dollars already spent in the traditional commercial television cable networks based on coaxial cables, and the fact that the network hardware costs would not be shared.

The third alternative, HFC, is widely recognized as the emerging standard for both cable and telephone companies as it allows one to use fiber-optic cables to carry their signals to large groups of homes (e.g., 500), called fiber service areas (FSAs), yet still use the existing coaxial cable to carry those signals into each home individually. Thanks to emerging coaxial cable systems, cable modem technology, and quadrature amplitude modulation (QAM) of digital information onto analog carriers, HFC will eventually allow the existing cable plant to be upgraded to digital transmission and serve 200- to 1,000-household neighborhoods with up to 500 MPEG-2 digital channels, in addition to the existing capacity of 65 analog channels [40], while leaving some bandwidth available to return, control, and personal communication channels. In the meantime, most cable TV systems, initially designed to pump analog data in one direction only, must be rebuilt to carry digital data in two directions. It is estimated that 25% of American homes will have upgraded cable systems by the end of 1998. It may be argued that bidirectional communication will cause congestion periods similar to those commonly-observed over the Internet when several interactive users transmit large data files simultaneously. Nevertheless, in interactive VoD, bidirectional communication is needed only for sporadic exchanges of short control messages between subscribers and service providers. In summary, while ADSL can be seen as an intermediate support solution, HFC is rapidly becoming suitable for VoD applications until we have full-optic fiber upgrades.

Even with vast improvements in technology, the architecture of the traditional television broadcast network should thus remain essentially the same, the broadcast signal being progressively supplanted by a digital signal. Similarly, the tuner currently used in CPEs to select a desired channel from the received signal will be upgraded to a miniature personal computer called a set-top box (STB). The STB is the bridge between the subscriber's display devices, peripherals, and input devices (such as hand-held infrared remote controller), and a communication channel, connecting the STB to the information infrastructure of service providers. It will typically include memory, an MPEG decoder chip, a cable modem, a processor for connection management, a frame buffer, and other "glue" logic [67]. The STB will be in charge of sending, receiving, and processing control messages and customer input, and receiving, decoding, and forwarding the decoded frame to the video display.

1.2 Research Objective

The operation of a VoD server comprises two distinct phases. First, upon receiving a request, the server must be able to reserve resources and decide if the request can be honored. This is the *program scheduling* phase, in which the VoD server uses an *admission control* algorithm to decide when to start program transmission, based on constraints such as the available capacity of the movie library on disks or disk arrays in a compressed format. Once a service has been scheduled, run-time operation of the VoD server will ensure continuous and seamless playout by retrieving in real-time and delivering a stream of compressed full-motion video data via the access network. In T-VoD, VCR actions after the start of a program also require specific mechanisms, which may involve both the local VoD server and CPE during the servicing phase.

Customers' QoS during scheduling and servicing phase is of paramount importance, given that economical viability and future large-scale deployment of VoD ultimately depend on customers' appreciation of the service delivered. Typical application QoS parameters for images and video include image size, frame rate, startup delay, synchronization between multiple streams, reliability and subjective factors such as the importance of the information to the user. Network QoS parameters include bandwidth, delay, jitter and loss rate. End-system parameters include CPU load, utilization, buffering mechanisms and storage related parameters. The focus of this dissertation is on the constraints imposed on resource allocation by QoS requirements at the user level. We approach this problem by focusing on the *interactions* between customers and the VoD server, which take place during the scheduling phase and in the case of T-VoD, upon request for VCR actions during the servicing phase. During the scheduling phase, customers' QoS can be expressed in terms of waiting time before receiving service, or *admission latency*, and the defection rate due to long waits. Also, when extra resources are added, video material changed, or in the event of disk failures, taking resources off-line for reconfiguration should incur minimum disruption of service to existing connections. Thus, customer's QoS is also affected by service *availability*, or minimum support provided by the VoD server in case of reconfiguration. In T-VoD, support for VCR actions provided to customers depends on the fraction of VCR actions which are (1) blocked; (2) promptly served in a continuous fashion, that is, a customer is able to fully control the duration of interaction; and (3) promptly served in a discontinuous fashion; discontinuous interactive functions can only be specified for durations that are integer multiples of a predetermined time increment. This classification indicates both customers' QoS received, as measured by their ability to obtain continuous service, and customers' *QoS degradation*, which can be either graceful (fraction of discontinuous actions) or not (fraction of blocked actions).

Several factors may affect customers' QoS in interactions. First of all, similarly to the current cable TV, VoD systems will experience variations in the request arrival pattern for videos on various timescales. For instance, one can observe daily variations in the request rate between "prime time" (e.g., circa 8 p.m.) and "off-hours" (e.g., early morning). On a larger time scale (e.g., one week), changes in movie popularities due to new releases or customers' loss of interest in current titles over time, may also cause changes in the request rate. Furthermore, customers' willingness to wait for service and the set of most frequently-requested movies may change over the course of a day. (For example, movies that appeal to children may register their peak access during the afternoon while other movies may be popular with adults in the evening.) Lastly, insertion of advertisements or movie previews may subsidize VoD service providers on a daily basis in the same way as traditional video and pay-per-view services. Such changes require storage reconfigurations, during which QoS degradation should be avoided. Our goal is to determine how to achieve *service scalability*, which is defined as *the ability to accommodate these possibly significant workload fluctuations and overloading situations without affecting customers' QoS in interactions.*

When designing a VoD server for scalability in interactions, one must deal with several constraints on resource allocation and usage. First, a VoD server has to offer a predetermined selection of movie titles, and support non-uniform access to these movie titles. Access locality in video services, caused by customers' specific demand patterns, is best described by a commonly-used criterion for partitioning movie titles, known as probability of movie access, or "popularity" [65]. Second, in addition to movie selection, the maximum number of concurrent channels and storage organization determine the cost of a VoD server. For a given cost range, the VoD server throughput, defined as the average number of customers served per movie transmission, should be maximized as it indicates the service provider's revenue. It is essential to satisfy these constraints while providing maximum QoS. The objective of this research is thus to specify long-term resource allocation and usage in VoD end-systems so that a large collection of movie titles and channels can be accessed by a maximum number of users with scalability in interactions, at the lowest cost.

1.3 Approach

In our study of service scalability in VoD end-systems, we considered the following three orthogonal issues: (1) how to dimension and organize storage for scalability, (2) how to improve scalability in D-VoD during the scheduling phase, and (3) how to improve scalability in T-VoD during both scheduling and servicing phases.

For a given movie selection and channel capacity, the storage organization of a VoD server will determine server cost — which depends on the required number of disks and amount of RAM needed for synchronous retrieval — and customers' QoS. Small-scale VoD servers which store one movie title per disk (OMPD) typically trade availability — disks can fail or be brought off-line for update without affecting the entire service — for increased latency and costly, inefficient, and unbalanced use of storage capacity. Unlike OMPD, VoD servers which store movie titles using various striping schemes [77] of videos across disks ensure a very low latency and high storage utilization. In this case, however, reconfigurations risk the availability of the entire VoD server. Ideally, a scalable storage architecture should combine the advantages of both schemes in order to be (1) versatile, that is, continuously available regardless of changes in the offered material, (2) expandable to ensure that new devices can be added easily, and (3) evenly accessed. Versatility and expandability are important properties since they define the ability to reconfigure the VoD server without disrupting service during the reconfigurations. In addition, access to storage devices should be as uniform as possible, in order to avoid "hot spots" and inefficient resource utilization. Thus, it is reasonable to expect that at the local level, the storage should consist of several clusters accessed with the same frequency, each cluster serving as a repository for a subset of all movie titles, for which it will provide a fraction of the total channel capacity. In both D-VoD and T-VoD, we therefore address how to choose clusters' capacity and how to assign videos to separate devices so as to maximize customers' QoS at minimal cost.

Once a storage organization has been specified, one possible way to improve scalability in D-VoD and reduce per-customer system cost is to delay the D-VoD server's response to requests *within customers' tolerance* and "batch" those requests made by many different viewers for the same movie within a *batching period* [10]. This method may potentially increase *throughput*, or the average number of service requests granted per program, by using multicast communication. From the service provider's perspective, this is a very attractive feature since the number of concurrent clients is not upper-bounded by resource availability. Furthermore, customers' average latency may also be reduced as requests for the same movie that arrived within the same batching period do not have to compete for resources. It should be noted that in T-VoD, such improvement in scalability is harder to achieve since the full support for continuous interactive functions requires dedicating resources to serve individual requests, which may degrade the system to a non-sharing mode.

The choice of a batching policy plays an important role in determining both customers' latency and system throughput, and therefore commercial success and immediate deployment of multicast VoD systems. We investigate a batching strategy which sources the same material at equally-spaced intervals, called *phase offsets*. This kind of VoD service, in which subscribers who order a particular movie to start within a specific time window are grouped together, is termed "Near-VoD" (NVoD) [12, 65]. It is becoming increasingly popular with the telecom, cable, broadcast, and content companies as it offers the potential to provide scalable, cost-effective digital media services. In addition to the improved scalability during the scheduling phase, the main advantage of NVoD systems over other batching policies is that, by keeping the batching interval nearly constant per movie title, it is possible to extend D-VoD service to provide customers with *limited and scalable VCR capability*. From a service provider's point of view, this feature is quite attractive for preliminary experimentation with interactive service, before making a large investment into T-VoD-like service. Indeed, limited continuity in VCR actions can be provided by caching a small amount of video data (e.g., 5 minutes' worth of video) in a buffer located in the CPE. This buffer can then be accessed without removing the customer from the multicast group. Moreover, staggered phase offsets support for discontinuous VCR actions is provided by allowing customers to specify the length of video they want to skip, possibly in integer multiples of the phase offset duration. In this case, the NVoD server will reassign the customer to the multicast group whose playout point is the closest to that requested by the customer. From a customer's perspective, another attractive feature of NVoD service is the ability to be informed exactly when the transmission will start. Thus, even when resources are scarce, predictable response times can be guaranteed for all incoming requests.

For a given channel capacity, movie selection, and storage organization, the number of channels allocated to each movie title depends on its popularity. For instance, in the case of a rarely-requested movie, the NVoD server will probably allocate very few channels to it so that more channels may be allocated to popular titles, and therefore, more customers may be served. Customers' willingness to wait for service will also impose constraints on the maximum acceptable phase offset, and hence on channel allocation. Based on these two conflicting constraints, we present heuristics to optimize NVoD server objectives, such as maximum throughput, and minimum phase offset, indicating customers' QoS.

In multicast D-VoD systems such as NVoD, scalability is primarily affected by the time-dependent request arrival pattern, and the storage organization of the VoD server, which may consist of one or several disk arrays. Ideally, a scalable batching policy should always prevent the degradation of customers' QoS and server throughput, and remain insensitive to fluctuations in the request rate. Performance should thus be as *self-similar* as possible, that is, remain approximately constant and acceptable on various timescales, regardless of the duration and amplitude of high load periods. If batching is done without regulation in the channel allocation process, extended periods of demand surge may cause congestion cycles, resulting in a scalability problem. As we shall see, this phenomenon is due to the inherent lack of variability in movie lengths, which makes uncontrolled channel allocation prone to channel clumping when periods of high loads alternate with periods of low traffic intensity. This problem is exacerbated by the heavily-localized access to these movies, and the disparity between the number of individual movie titles usually held in most storage organizations and the much higher concurrent channel capacity allocated to these movies. We, therefore, address next the problem of identifying and evaluating scalable batching policies under nonstationary load conditions. In parallel with this study, we propose scalable alternatives to NVoD, which, unlike the latter, are applicable to a wide range of clustered storage organizations, and may thus offer a better tradeoff between customers' QoS, and server's throughput.

The last issue addressed in this dissertation is how to provide a T-VoD-like fully-interactive on-demand service in multicast VoD systems without sacrificing service scalability. In the absence of adequate mechanisms for using VoD system resources other than simply sourcing video materials in playback mode, continuity in VCR actions can only be provided intermittently at best (e.g., in NVoD). To alleviate this limitation, we propose mechanisms for unrestricted VCR functionality with minimal degradation of system scalability, and therefore economical viability, and evaluate them with scalable batching policies. The basic idea behind these mechanisms consists in using a pool of "interaction" channels (or I-channels) in conjunction with partial caching of programs in the CPE buffer, to execute operations which would otherwise be blocked. The low-latency buffer is then actively employed to *reclaim dedicated resources* by merging the customer back in synchronization with a "batching" channel (B-channel). This is simply performed by prefetching frames or groups of frames while an I-channel is serving the customer in playback mode. This way, both the VoD server and the CPE buffer can work synergistically to decrease the probability of blocking VCR actions while preserving the service scalability achieved by the VoD server.

In such interactive multicast VoD systems, both customers' QoS and VoD server scalability are affected by a combination of four independent factors: (1) the CPE-buffer size and management; (2) the ratio of the number of I-channels to that of B-channels; (3) customers' request and interaction behavior; and (4) the VoD server's batching policy. A larger CPE-buffer size is preferable to reduce the load on the pool of I-channels by making a larger portion of the video program available for immediate access and by making it more likely for an I-channel customer to find and join a target multicast group after a VCR action. However, mechanisms to protect intellectual property rights — so that service providers are able to maintain control of their data and thus are able to stay in business — and affordability constraints will impose an upper limit on the buffer size, which will most likely be restricted to a few minutes' worth of video. It is therefore necessary to find the appropriate batching policy and partition between B- and I- channels in order to provide a viable tradeoff among server throughput, the admission latency experienced by new requests, and the blocking probability of VCR actions. Note that the tradeoff between admission and interactions is less clearcut as it may first appear, since more channels allocated to a frequently-requested movie will have the two opposite effects of (1) admitting more customers into the system, thus increasing the load on I-channels; (2) making merge attempts from I- to B- channels for that particular movie title more likely to succeed, thus reclaiming resources more efficiently. To investigate the complex interactions among the above-mentioned factors, we evaluate support for VCR actions and the VoD server's batching efficiency by considering various realistic scenarios of customer interactive behavior, including parameters such as the level of interactivity or the duration of VCR actions.

In summary, our three-fold approach to service scalability in VoD end-systems looks at storage organizations for low latency and high service availability, request batching for high throughput, and synergistic collaboration between VoD server and CPE buffer for full and scalable VCR functionality.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 addresses the problem of organizing resources, so as to make a large collection of movie titles and concurrent channels available at the lowest possible cost, while offering the highest admission QoS. The coarse-grained striping (CGS) scheme in disk arrays is shown to be a potential candidate for its cost-effective storage usage and low admission latency. We study the feasibility of storage organizations in which disks are partitioned into several clusters to reduce service disruptions during reconfigurations. In parallel with this, we present an algorithm for optimal video allocation across clusters, and show that high service availability can be achieved in a cost-effective way with minor side-effects on customers' admission latency. We extend these results to show that full-fledged VCR functionality can be provided at a reasonable cost.

In Chapter 3, we present an analytical (in contrast to commonly-used simulations) approach to program scheduling in near video-on-demand (NVoD) systems. The proposed approach to analytical modeling integrates both customers' and service provider's views to account for the tradeoff between system throughput and customers' partial patience. We first determine the optimal scheduling of movies of different popularities for maximum throughput and the lowest average phase offset. Next, we deal with a variation of NVoD called quasi video-on-demand (QVoD), in which programs are scheduled based on a threshold on the number of pending requests. The throughput is found to be usually greater in QVoD than in NVoD, except for the extreme case of nonstationary request arrival rates.

This observation is then used to improve throughput without compromising customers' QoS in terms of average phase offset and the corresponding dispersion.

Chapter 4 presents an comparative study of various batching policies in D-VoD servers under realistic conditions including customers' behavior, storage organization, and time variations of request arrival rates. We focus on disk-array-based D-VoD servers, whose movie selection and channel capacity are determined by cost and the underlying striping scheme. We analyze the well-known lack of scalability of on-demand batching, and introduce a methodology to measure scalability. We then compare various scalable batching policies and show that even the simplest batching policy may vastly improve on-demand channel allocation. Finally, we study the feasibility of scalable batching in clustered disk-arrays, and show that high service availability and scalable batching can be achieved in a cost-effective way.

In Chapter 5, we propose and evaluate a framework for fully-interactive, yet scalable, on-demand service in multicast VoD systems. As the first step, we present support for discontinuous and intermittently-continuous VCR actions, the playback (default) mode of a multicast VoD system. Next, we show that unrestricted support for interactive operations can be provided by allocating a portion of the VoD-server's channel capacity for interactive operations which would otherwise be blocked. These interaction channels are used in conjunction with partial caching of programs in CPEs to make sharing and scalability transparent to the viewers. The proposed framework for full VCR functionality is applied to the various scalable batching policies that have been presented in Chapter 3 and evaluated in Chapter 4. For the sake of realism, we introduce an idealized, yet realistic, model of customers' admission and interaction behavior which captures several key features of customers' demands for interaction support. We use this model to show that batching policies under the proposed framework can make an attractive tradeoff between system scalability and viewers' VCR functionality. We also show, for various CPE-buffer sizes and channel capacities, that our framework for interactions provides an effective support for graceful QoS degradation in case of resource shortage. Finally, we identify through simulation that active CPE-buffer management improves customers' QoS when interaction behavior is biased towards a particular type of VCR action.

This dissertation concludes with Chapter 6, which summarizes our contributions and suggests future work.

CHAPTER 2

RESOURCE ALLOCATION
IN VIDEO-ON-DEMAND SERVERS

2.1 Introduction

Our objective in this chapter is to specify storage organization and long-term resource allocation in distributed VoD servers so that a large collection of movie titles and channels can be accessed with the highest QoS and at the lowest cost. In case of distributed VoD service, customers typically interact locally with front-end VoD servers via an access network. In the rare case of requests for movie titles not available locally, nearby local VoD servers can be queried. Alternatively, data can be transferred at high speeds from the closest archive and cached locally for playout. For the most popular movies, customers' QoS during the scheduling phase is therefore affected by the architectural decisions for the choice of a storage medium and resource allocation in local VoD servers.

We saw in Chapter 1 that local VoD servers (D-VoD or T-VoD) are responsible for billing and connection management during the scheduling phase, and real-time retrieval and delivery of digital bit streams containing compressed full-motion video data during the servicing phase. In this chapter, we focus on the program scheduling phase, in which the VoD server decides when to start program transmission based on the available capacity of the storage system. During the scheduling phase, customers' QoS can be expressed in terms of waiting time before receiving service, or *admission latency*, and minimum support provided by the VoD server in case of reconfiguration, or *service availability*. In the first case, long waits may cause a high defection rate and loss of viewers. Also, when extra resources are added, video material changed, or in the event of disk failures, taking resources off-line for reconfiguration should incur minimum disruption of service to existing connections. From the VoD server's point of view, the ideal storage architecture can thus be seen as a combination of (1) *scalability*, defined as the ability to accommodate significant workload fluctuations and overloads without affecting admission latency; and (2) *versatility*, defined as the ability to reconfigure the VoD server with minimal disturbance to service availability during these changes. A high level of versatility is also desirable for *expandability*, to ensure that new devices can be added easily.

At the local level, the video server requires a storage medium which offers fast access, has a large capacity, is relatively inexpensive, and is capable of the random access needed for VoD operation. Tertiary storage technologies such as magnetic tapes, optical jukeboxes, or the new Digital Versatile Disc (DVD) jukeboxes are highly cost-effective because they provide large storage capacities at low cost. However, their random access time is slow, and their throughput is low. Consequently, they are better suited to movie archives [26], providing a large selection of movie titles that is not frequently accessed, and for which synchronous retrieval of video data is not as critical as at the local level. On the other hand, due to the voluminous nature of video data (e.g., a 100-minute MPEG-1 movie requires 1.125 GB), and the high cost of RAM ($40/MB in 1996), storing videos entirely in RAM is prohibitively expensive. It is therefore commonly accepted [26, 71, 77] that a video server is more cost-effective if it relies on magnetic disks instead of RAM for the storage and synchronous retrieval of the most popular videos. A small, appropriately chosen amount of RAM is then used to buffer retrieved video data during each disk access, in order to compensate for the relatively high latency for data access (e.g., 10-20 ms) and for the disk access rates (e.g., 30-60 Mbps), typically greater than compressed video data rates. Note that using magnetic disks as the sole storage medium may become too expensive if the number of movies accessible at the local level is too large. In this case, local storage can be organized as a hierarchy that combines the cost-effectiveness of tertiary storage to store the "least-frequently accessed" popular movies, and the high performance of fixed magnetic disks.

2.1.1 Approach

When designing a VoD server, one must deal with several constraints on resource allocation to provide scalability, versatility, and ensure load to be evenly balanced. First, a VoD server has to offer a pre-determined selection of movie titles, and support non-uniform access to these movie titles of heterogeneous popularity. In addition to movie selection, the maximum number of concurrent synchronous channels and storage organization — e.g., schemes for laying out the videos on multiple disks for efficient disk bandwidth utilization —

also determine the number of magnetic disks and amount of RAM needed for synchronous retrieval, which in turn determine the cost of a VoD server. For illustration of these complex interactions, we will first overview two basic, well-understood types of storage technology, specifically chosen for their distinct ways to allocate resources among different movie titles, and to illustrate their respective limitations. The first type "completely" partitions the storage among different movie titles. An example of such organizations, called *completely-partitioned* (CP) organizations, may be found in small-scale VoD servers which store one movie title per disk (OMPD). The second type completely shares the storage among different movie titles. An example of such partitions, called *completely-shared* (CS) organizations, is VoD servers which store movie titles using fine-grained striping (FGS) or coarse-grained striping (CGS) [77] of videos across disks in order to effectively utilize disk bandwidth.

Both forms of storage constitute, to some extent, the two extreme cases. CP organizations typically trade availability — disks can fail or be brought off-line for update without affecting the entire service — for increased latency and costly, inefficient use of storage capacity. CS organizations, on the other hand, ensure a very low latency and high storage utilization, but reconfigurations risk the availability of the entire VoD server. Thus, it is reasonable to expect that at the local level, the storage consists of several clusters and combines both storage models into a hybrid organization. Each cluster will then serve as a repository for a subset of all movie titles, for which it will provide a fraction of the total channel capacity. In this chapter, we shall therefore address how to partition a CS organization into clusters, how to choose clusters' capacity, and how to assign videos to separate devices so that customers' QoS may be maximized at minimal cost.

The remainder of this chapter is organized as follows. We first focus on D-VoD servers because, while being the easiest system to understand and the first implemented in research labs, deterministic service is *the minimum service one can expect from a video server*. As background, we present the two basic storage organizations and compare OMPD, FGS, and CGS in terms of admission latency and cost to provide a given channel capacity and movie selection. We next investigate the increase in cost and customers' latency that enables disk arrays to be clustered to provide customers with higher service availability. Given that VoD service providers target at the home consumer market, we study resource allocation in various realistic scenarios including time variations in the request arrival pattern. We show that OMPD, although often quoted as a naive approach to storage configuration, may be economically acceptable for highly reliable service of a small number of movies to a large user population. Lastly, since customers' interactive behavior after admission affects the video retrieval rate and the holding time of VoD server's resources, we adapt our methodology to T-VoD service. This chapter concludes with Section 2.7.

2.2 Two Basic Storage Organizations

2.2.1 Optimal Concurrent Access Profile in CP Organizations

CP organizations completely partition the storage among different movie titles by storing one movie per disk (OMPD). While not extremely attractive in terms of cost compared to other sophisticated storage technologies, OMPD has definite advantages such as reliability, since when a disk fails, only one copy of the movie becomes unavailable. Currently, a disk has capacity to hold 1 or 2 movies in compressed format ($1-9$ GB), while it has the throughput to allow 5 to 20 viewers to simultaneously access its contents ($30-60$ Mbs). Thus, if a video is not replicated on several disks and entirely stored on a single disk, the number of concurrent video streams that can be supported is bounded above by the disk bandwidth. The popularities will therefore dictate the number of disks that should be allocated to each individual movie title, hence the partition of the concurrent channel capacity among movie titles. (A more popular movie will be replicated and stored on multiple disks.)

Let d denote the number of disks to be partitioned among N movie titles, R the bandwidth required per stream, B the bandwidth of a single disk in number of video streams, S the storage required per movie, and $C_d > S$ the capacity of a single disk. For a storage organization supporting N different movies, popularity can be defined as a vector of access probability $[p_1, p_2, \cdots, p_N]$, $\sum_{m=1}^{N} p_m = 1$, where p_m is the probability that a client will request movie m. Thus, if the aggregated request arrival process is Poisson with parameter λ (or $\lambda(t)$ in case of time dependence) — as has traditionally been used in telephony — then, according to the well-known result of splitting a Poisson process [21], the request arrivals at movie title m are a Poisson process with parameter $\lambda_m = p_m \lambda$. The traffic intensity is given by $\rho = \frac{\lambda}{s\mu}$.

Let's consider the optimal partitioning of the disk capacity of a VoD server among movie titles requiring an identical playout rate. A reasonable partitioning policy is the proportional allocation, which attempts to allocate $d_m = p_m d$ disks to movie title m. However, as $p_m d$ is usually not an integer, the video server may be forced to allocate $d_m = \lceil p_m d \rceil$ for $m > 1$ and $d_1 = d - \sum_{m=1}^{N} d_m$. The major problem with this "blocking effect" is that the *maximum sustainable traffic intensity* — which is the maximum value of $\rho = \frac{\lambda}{s\mu}$ sustainable while keeping the system stable — will be less than 1, as the proportional allocation policy may lead to a traffic intensity $\rho_m \geq 1$ for movie title m. In order to reduce the blocking effect, it is possible to specify an *optimal concurrent access profile* (OCAP), which is the partition $[d_1, \cdots, d_N]$ of d disks among N movies so that minimizes the customers' average waiting time. OCAP specification determines, for a given traffic intensity, all the stable partitions, and selects the one corresponding to the highest maximum sustainable traffic intensity.

Our approach in calculating the waiting time is to model the partitioning of VoD storage for a set of movie titles, as a group of N parallel queueing systems of type $M/D/c^1$, denoted by $M/D/d_m.B, m = 1, \cdots, N$. If the VoD server does not support any interactivity, the service will be provided deterministically at rate $\mu = 1/L$, where L is the average length of movies. For simplicity, we assume that all movie titles are of an identical length, since most movie titles are known to have playback lengths of about 90 – 100 minutes. While a Gaussian or uniform distribution between, say, one and two hours might provide a better match, it will not affect the general conclusions drawn from our analysis. Denoting by $EW(M_{\lambda_m}/D/d_m.B)$ the average waiting time for each virtual queueing system $M_{\lambda_m}/D/d_m.B$ of type $M/D/c$ with Poisson arrivals at rate λ_m, deterministic service, and $d_m.B$ servers, our optimization problem can be stated as:

Problem VoD OCAP: Given d concurrent disks, $\vec{d} = [d_1, \cdots, d_N]$ partition of the disks among $N > d$ different movie titles or groups of movie titles, λ the aggregated request arrival rate, μ the average service rate, $[p_1, \cdots, p_N]$ the movie popularity vector and $\lambda_m = p_m\lambda$ the arrival rate at each $M/D/d_m.B$ queue corresponding to movie title m, determine

$$\min_{\vec{s}} \sum_{m=1}^{N} p_m EW(M_{\lambda_m}/D/d_m.B) \quad \text{such that} \quad \sum_{m=1}^{N} d_m = d.$$

This optimization problem can, in general, be solved using an efficient enumeration technique as we showed in Appendix A. Also, $EW(M_{\lambda_m}/D/d_m.B)$ can be accurately inferred from the corresponding $M_{\lambda_m}/M/d_m.B$ queueing system with exponentially-distributed service times by using a technique similar to the one outlined in [56].

2.2.2 CS Storage Organizations

CS storage organizations completely share the storage among different movie titles by striping videos across *disk farms* or *disk arrays* in order to effectively utilize disk bandwidth. Unlike OMPD, in which the maximum number of concurrent channels for a particular movie title is determined by the number of disks allocated to that particular movie title,

[1] This notation reflects standard queueing theory nomenclature [17, 54], whereby the first letter M stands for a Poisson arrival process, the second letter D indicates that the probability distribution of the service times is deterministic, and the last number corresponds to the number of servers.

in CS organizations each disk can devote its transfer capability to playing whichever movie has been requested. Thus, unwatched movies require disk capacity but do not waste disk bandwidth, and the concurrent channel capacity is fully available to any movie title. In this chapter, we consider striping over disk arrays which, with specialized software and/or hardware controllers, distribute consecutive logical data units on consecutive disks in a round-robin fashion. Two popular schemes are presented in [77]. In *fine-grained striping* (FGS) (similar to RAID-3), the stripe unit is relatively small (e.g., a byte) and every retrieval involves all d disks that behave like a single *logical* disk with bandwidth dB. In coarse-grained striping (CGS), each retrieval block consists of a large stripe unit which is read from only a single disk, but different disks can simultaneously handle independent requests. CGS with parity information maintained on one or several dedicated disks corresponds to RAID-5 [19, 76].

We shall henceforth restrict our study to constant bit rate (CBR) media data whose bandwidth requirement remains fixed throughout the entire display of a video. In both striping schemes, due to the periodic nature of video clips, data for videos in service is retrieved in fixed-length "rounds," using the C-SCAN seek optimizing disk scheduling algorithm [77]. C-SCAN eliminates random seeks to arbitrary locations by sorting videos and traversing disk surface from the innermost pending request to the outermost pending request. In disk-array-based VoD servers, new customers are served by transferring their requests from a request list to a service list. Unlike FGS, in which only one service list is needed, in CGS a service list is maintained for each disk. If the number of stripe units per video is a multiple of the number of disks[2] the storage server operation at the end of each round is to set the service list for every disk to that of its preceding disk. Otherwise, the *current available bandwidth* scheme presented in [77] is used for starvation-free disk bandwidth allocation. New customers are served by transferring their requests from a request list to the service list of the first disk. For each video, consecutive stripe units of size corresponding to the duration of a round are stored on the d disks in round-robin fashion. Retrieval of each video stream then proceeds in lock-step, that is, each client accesses exactly one disk during each round, and consecutive disks are accessed by the same set of clients during successive rounds. The more general issue of accessing variable bit rate (VBR) encoded media stream using CGS is beyond the scope of this dissertation. Generally speaking, efficient storage

[2] This can be enforced by appending movie previews or advertisements. Such content insertion practice may subsidize VoD service providers in the same way as traditional video and pay-per-view services. As we shall see, however, the stripe unit and number of disks needed for a 1,000- or 2,000-channel CGS disk array are so small that typically less than a minute worth of content insertion can be expected.

utilization while guaranteeing hiccup-free continuous display depends on the elected block placement policy and on load balancing techniques [57]. Opting for variable-size blocks corresponding to video segments of constant duration is shown in [24] to be cost-effective. It is also well-suited for a read-only environment such as VoD, since lock-step retrieval is made possible. Whatever the case may be, the schemes presented in this paper can deal with VBR data by using standard techniques such as pushing data at constant rate R and allocating sufficient buffer space at the client to smooth variations with respect to R.

Differences in degree of concurrency and parallelism affect storage capacity, and therefore, performance modeling and cost evaluation. A high degree of parallelism in FGS eliminates the need for replication, as conjectured in [65], but at the cost of d times higher latency overhead per data access. In CGS, the number of concurrent accesses to a particular movie title is limited by the load on the disk holding the first video segment, but high-degree concurrency is better adapted to retrieval of material such as videos, with relatively low bandwidth in comparison with the bandwidth of a single disk (each disk having enough bandwidth to serve several videos). Consequently, it is proved in [77] that the retrieval cycle is much lower in CGS than in FGS (e.g., 1 second instead of 12 in a disk array supporting 1,000 channels), thus incurring lower disk and RAM requirements for continuous data retrieval and transmission. In addition to distributing the workload uniformly across disks, disk striping also enables multiple concurrent streams of a video to be supported without replicating the video. With a large number of disks (e.g., 1,000), however, CGS has been documented in [26] to cause substantial admission wait if the system is highly-loaded, unless movies are replicated in proportion to popularities. Nevertheless, this is an extreme case not representative of local VoD service.

2.3 Choosing a Storage Organization

2.3.1 Cost and Storage Capacity

In our evaluation study, and for the remainder of this dissertation, disk parameters such as settle time, worst-case seek time, worst-case latency, rotational latency, cost and capacity are drawn from a commercially-available disk [103]. We also assume that all movie titles are compressed with MPEG-1 and have the same length (=100 minutes), as most movie titles are known to have playback lengths of about 90 – 100 minutes. While a Gaussian or a uniform distribution between, say, one and two hours might provide a better match, it does not change the general conclusions of our analysis. With an inner track transfer rate of 45

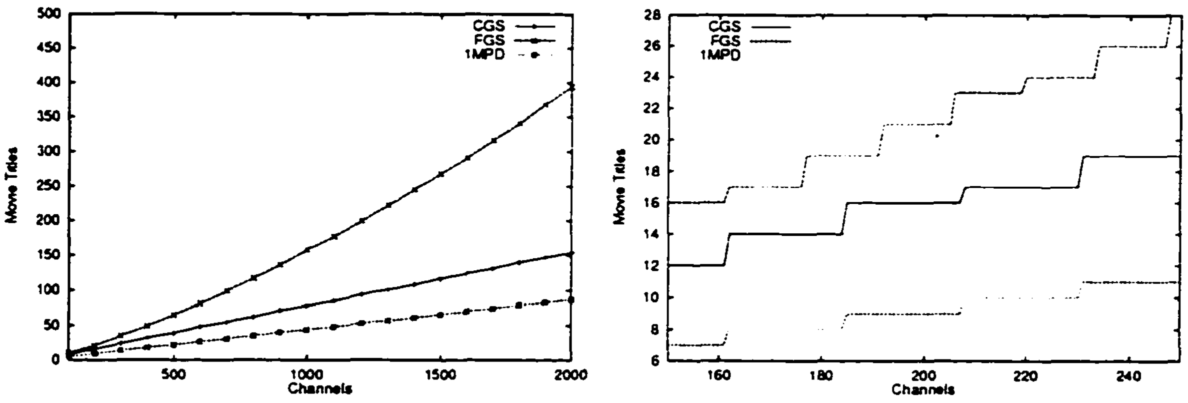


Figure 2.1: Number of movie titles supported (and close-up).

Mbps and a capacity of 2 GB, each disk is able to support several MPEG-1 channels at 1.5 Mbps, whereas it can hold only one 100-minute movie (1.125 GB). This disparity is even greater with higher quality MPEG-2 movies: a 90-minute movie would need about 5 GB of storage and an I/O bandwidth of 8 Mbps. In both FGS and CGS, analytical techniques elaborated on in [77] are used to determine the optimal retrieval cycle of a given number of concurrent channels at minimal cost, based on (1) the latency for accessing data on disk, which is the sum of seek, rotational, and settle latencies[3]; (2) round-robin placement of consecutive stripe units; (3) C-SCAN disk scheduling; and (4) number of disks and amount of RAM needed for timely frame retrieval at playback rate R (and therefore, hiccup-free continuous display). Cost evaluation is based on (1) the amount of RAM needed to provide two-stage buffering at each channel in a dual buffer system; and (2) the number of disks needed to provide the required transfer rate. For illustration purposes, we assumed storage costs in 1996, that is, $40/MB for RAM, and $1500 per individual disk of capacity 2 GB [77].

Figure 2.1 plots the relationship between maximum movie selection and the channel capacity supported by each scheme. The number of disks was assumed to be the same in OMPD and CGS. As can be deduced from the piecewise-constant curves of the second graph of Figure 2.1, the same number of disks can be used to support a certain range of channel capacity, depending on the amount of RAM available. In case of FGS, the penalty caused by accessing all disks in parallel for each block retrieval requires more disks, and hence more RAM, to support the same channel capacity as in CGS and OMPD. Consequently, even though a larger movie selection can be stored, the storage cost in FGS increases almost

[3] Parameters usually provided by the manufacturer (cf. [103]).

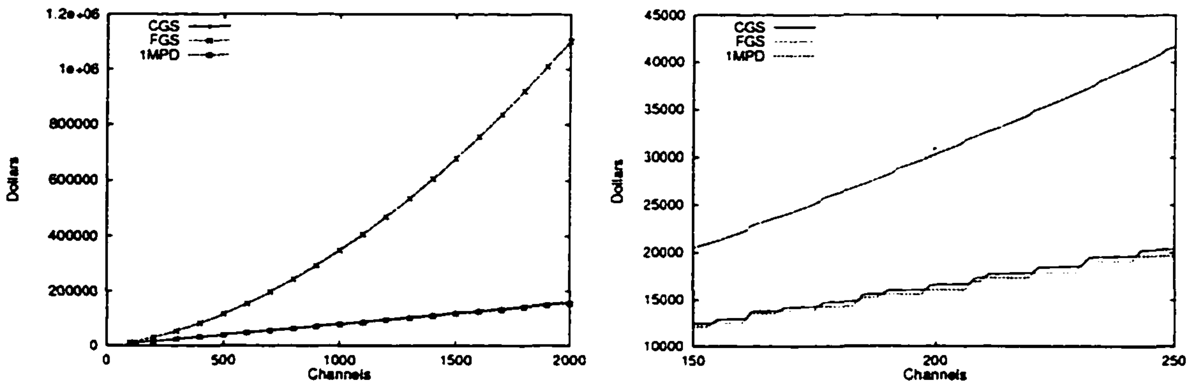


Figure 2.2: Storage cost (and close-up).

exponentially with the channel capacity, as shown in Figure 2.2. For the same channel capacity as in FGS, CGS and OMPD incur a much lower storage cost, which varies quasi-linearly with the channel capacity. Note that the storage cost is also slightly lower in OMPD than in CGS due to its lower RAM requirement. It is clear, however, that storing 1.78 times more movies in CGS is worth a marginal increase in cost with OMPD. Lastly, it is also interesting to note that only 44 disks are needed to support a 1,000-channel CGS disk array. For a stripe unit corresponding to one minute worth of video, this means that at most 22 seconds of content insertion are needed in order to ensure that the number of stripe units per video is a multiple of the number of disks, and therefore trivial, starvation-free request scheduling.

Ideally, the storage organization of a VoD server is chosen so a large collection of movie titles and concurrent channels can be made available with the highest QoS at the lowest possible cost. In reality, Figure 2.1 indicates that a feasibility region must be determined to satisfy both constraints as closely as possible. This issue is made clearer in two example constraints 1 and 2[4] in the first graph of Figure 2.3. In case of constraint 1, CGS yields extra storage space available for new video material or mirroring of existing movies. On the other hand, storing the required number of movies with FGS or OMPD is possible only if more channels than initially required are provided. As can be seen in the second graph of Figure 2.3, cost ultimately determines the choice of a system. In fact, regardless of whether the system is disk-arm bound (areas A and C) or disk-space bound (areas B and D), CGS can be shown to be the most advantageous scheme since (1) the difference in cost between FGS and CGS increases almost twice faster than the difference in number of movie titles

[4] For simplicity, the curves corresponding to each scheme are approximated, but this does not affect our conclusion.

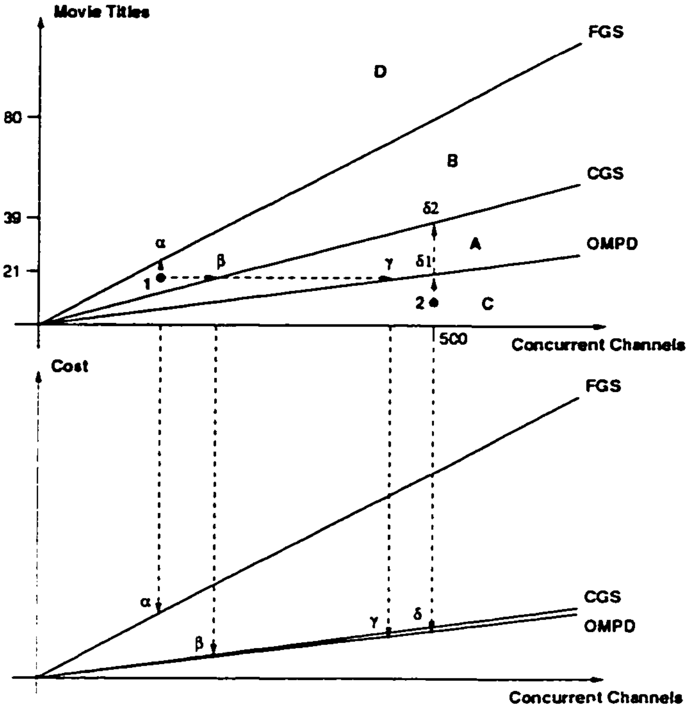


Figure 2.3: Choice of a storage organization.

supported; (2) at approximately identical costs and channel capacity, CGS can support more movie titles than OMPD. Similar observations can be made by considering storage of 90 – 100-minute MPEG-2 movies on Seagate's newer 9-GB *Ultra*$^{\text{TM}}$ drives (rather than 2 GB), whose transfer rate is three times that of older 2-GB drives.

Note that constraints located in areas B and D (respectively, constraints of *type* B and D) are not economically rewarding since they correspond to vastly under-utilized storage-limited systems which provide a large selection of movie titles to a small user population. This type of constraint is characteristic of movie archives, for which tertiary storage technology may be better suited. Inversely, it is economically more viable for companies to design front-end servers of type A or C and serve the most frequently-requested movie titles to a subset of the total viewer population. In the extreme case 2 of area C, a large number of channels is to be allocated to a small number of movies. Although CGS can support more movies than required, OMPD is still acceptable in terms of cost. The choice of a scheme will then depend on customers' QoS, discussed next.

2.3.2 Customers' QoS

2.3.2.1 Admission Latency

The first component of customers' QoS we investigated is the waiting time (in receiving the requested service) averaged over all movie titles. In a similar study in [55, 68], the overall blocking probability was chosen instead of the average waiting time. However, this choice is not sufficient to evaluate customers' QoS in VoD, because the blocking probability averaged over several queueing systems doesn't indicate whether customers requesting some arbitrary movie will be blocked with a probability close to 1, hence suffering an unacceptable long wait. We considered a capacity of 500 channels and 10 different movie titles. For these values, CGS and OMPD require 21 disks. To evaluate the gain by employing the OCAP in OMPD, we considered two cases of disk partitioning: (1) the OCAP EW-OPT, determined by the optimal heuristic in Appendix A, (2) the CAP EW-PROP, obtained by allocating the number of channels per movie in proportion to popularities. We adopt Zipf's law [10, 26, 102] as the stationary model of movie popularity, which has proven to work even when the viewing probability of videos is not stationary [18]. Zipf's law, generally recognized as representative of video popularities, was originally used for deriving the relationship between the number of references made to a word in a given text, and its rank order based on the same measurement. Zipf-like distributions are frequently used in social sciences to express the probability of selecting a particular object from a fixed number of objects where there is a skew towards some of the objects. In a Zipf-like distribution, we have:

$$p_m = \frac{f_m}{\sum_{j=1}^{N} f_j} \tag{2.1}$$

where the frequencies $f_m = \frac{1}{m^{1-\theta}}, m = 1, \cdots, N$, and θ specifies the skew: if $\theta = 0$ the distribution is pure Zipf; if $\theta = 1$ the distribution is uniform. The authors of [10] reported that $\theta = 0.271$ closely matches the popularities generally observed in video store rentals.

Simulation results, plotted in Figure 2.4, show that the admission latency is usually low for the traffic intensities that can be supported. The differences between the various schemes appear to be mostly in the maximum sustainable traffic intensity, beyond which the load exceeds the storage bandwidth and users have to wait for a long time before receiving service. This is because those channels that have been assigned to customers remain allocated for the entire length of a movie. As expected, FGS and CGS perform comparably since CS organizations provide customers with very low latency. For very high traffic intensities, however, higher concurrency in CGS is preferable. Separate simulations confirmed that

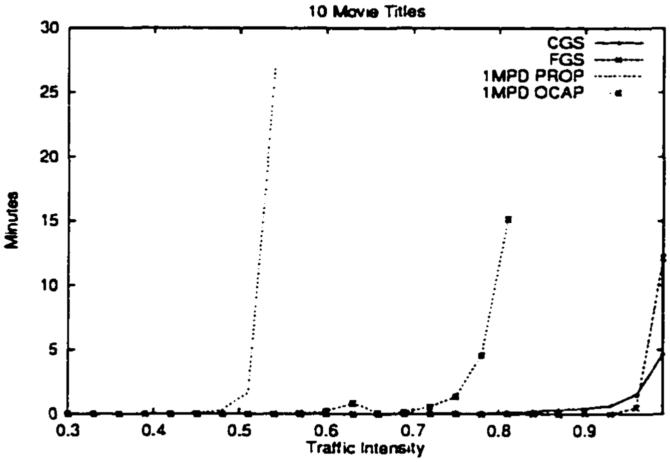


Figure 2.4: Average latency before service.

replication in CGS can only marginally affect customers' latency for a relatively small number of disks (typically, less than 100). In OMPD, the admission latency depends directly on the number of channels allocated to each movie title, that is, on the partitioning decisions and storage allocations made by the D-VoD server based on movie popularities. In this case, partitioning disks among movie titles according to EW-OPT performs better than EW-PROP.

The admission latency in CS organizations FGS and CGS depends only on the channel capacity regardless of the number of stored movie titles. In OMPD, the admission latency decreases with the selection of movie titles. In the extreme case of only one movie title, the latency in OMPD and CGS is almost the same and OMPD is better for availability. Storing *d* movie titles with OMPD, on the other hand, results in a very unstable system which cannot honor requests for most popular movies, unless more disks are added. In other cases such as 10 movie titles, the choice of a scheme depends on which is valued most, low customers' latency in CGS, or availability in OMPD.

2.3.2.2 Availability

Service availability in a VoD server can be measured by the minimum support provided in case of reconfiguration. In a sense, availability also indicates the VoD server's versatility, i.e., the ability to be reconfigured with minimal effect on service availability. A reconfiguration typically occurs (1) after a disk failure, or (2) when video material is updated, or (3) when the system capacity is upgraded. In the event of disk failures, a disk array can be made as reliable as OMPD by adding redundant data on one or several extra parity disks. In this way, if one disk fails, sophisticated algorithms [76] can be used to reconstruct data from the remaining disks, as a background process, while maintaining the same service level, at the cost of extra buffer. At present, the mean time to failure (MTTF) of a single disk is in the order of 600,000 hours (approximately 70 years) [76, 101]. Considering the fact that only 78 disks are needed to support 1,000 MPEG-1 channels using CGS, local front-end servers will typically contain less than 100 disks, and it is therefore realistic to assume an aggregate MTTF of 6,000 hours (approximately 250 days).

This failure rate is negligible in comparison with the much higher frequency of the other two types of reconfiguration, which risk the availability of the entire array by requiring all data to be reassigned to disks. Since VoD service will compete with — and eventually replace — the existing video rental facilities, the most popular movies held by front-end VoD servers will be renewed on a weekly basis. Furthermore, insertion of advertisements or movie previews may subsidize VoD service providers in the same way as traditional video and pay-per-view services. Such insertion may occur on a daily basis. When data is striped across all disks, adding disks also requires all data to be reassigned to the disks. Finally, gradually extending a video server may lead to a heterogeneous system, due, for instance, to disk models being discontinued [101]. In all these cases, as advocated in [32], high availability is achievable only by using disk arrays with *narrow striping* across multiple independent striping groups, or *clusters*.

2.4 Partitioning a Disk Array

Monolithic servers lack availability during rebuild operations and are not easily scalable to a wide range of requirements. These limitations led researchers to experiment with partitioned (or clustered) disk arrays for their cost-effective combination of concurrency, storage efficiency, and low customer latency. Depending on the total number of striping groups, however, reconfigurations and additional scheduling complexity during reconstruction may degrade performance for an unpredictable period of time, even when sophisticated replication or duplexing algorithms are used [38, 57]. In this section, we determine the tradeoff among cost, customers' latency, and availability, incurred by partitioning a disk array. The quasi-linearity of the cost functions in Figure 2.2 suggests that storage fragmentation in CGS may be done at near constant cost and capacity. However, partitioning a storage organization based on FGS, as recently proposed in [96], is not as economically appealing, due to the convexity of its cost function. So, we will restrict our study to CGS and assume that storage is partitioned into equal-size clusters. To achieve the appearance of a single resource from multiple clusters, we also assume that a storage switch is inserted between storage and clients. The more general issue of unequal-size clusters is left as future work. However, versatility is difficult to achieve with heterogeneous configurations, because they complicate (1) assignment of videos to clusters, and (2) video migration among clusters, both of which are needed to maintain a balanced. efficient system aş movie popularities change.

2.4.1 Eligible Configurations

Let's consider a 500-channel disk array with CGS, thus comprising 21 2-GB disks of the type considered in Section 2.3.1, and supporting a maximum of 39 movie titles of 100 minutes each (cf. Figure 2.3). Assuming that less than 39 MPEG-1 movie titles have to be stored, partitioning these 21 disks while keeping the cost as constant as possible leads to only two *eligible configurations*: 3 clusters of 7 disks each, or 7 clusters of 3 disks each. This example shows that partitioning a disk array to keep the total cost constant yields too few eligible configurations, hence too few alternative choices for an adequate combination of cost, capacity, and customers' QoS. If, on the other hand, the storage cost is allowed to vary within a given range, corresponding, for instance, to the cost of 1 or 2 disks and the RAM needed for full utilization, more configurations may become eligible and offer a better tradeoff. This is the case, for instance, if a cost increase as low as 5% is acceptable when a 500-channel disk array is to be partitioned.

Depending on the required channel capacity and movie selection, one may distinguish two cases. First, if these constraints define a point located in area A or C of Figure 2.3 (respectively, constraints of *type* A or C), the storage capacity is greater than the movie selection to be supported, hence leaving room for replicating the most frequently-requested movie titles. Such constraints naturally favor partitioning because, as we shall see, a small degree of replication can significantly reduce loss of viewers due to reconfigurations, even when the system is heavily-loaded. Moreover, movie titles can be replicated or added at will, without having to change the storage configuration. In Figure 2.5, variations of the total cost, number of movie titles, and number of disks, found analytically, are represented as a function of the number C of clusters, assuming a minimum channel capacity of 500. To make the graph more readable, various constraints on the number of movie titles are also indicated: (1) 39 movie titles, corresponding to the maximum number of movies stored on one CGS array that supports 500 channels; (2) 30 movie titles, corresponding to type A

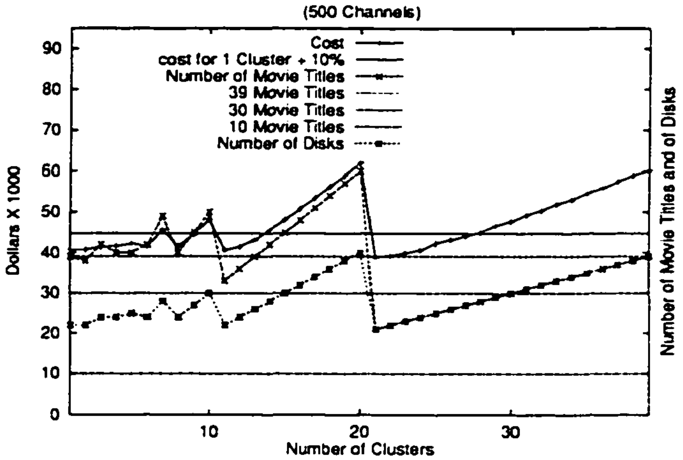


Figure 2.5: Storage cost, number of movies titles supported and number of disks for various number of clusters: constraints of type A and C.

constraint (for the remainder of the dissertation, we shall refer to (500 channels, 30 movie titles) as constraint A); and (3) 10 movie titles, which are a constraint of type C (we shall refer to (500 channels, 10 movie titles) as constraint C). We also indicated the maximum cost allowed, arbitrarily set to the cost of a 1-cluster system *plus* 10%. C was varied from 1 to 39.

As can be seen from Figure 2.5, increasing C from 11 to 20 results in a linear increase in cost, and correspondingly, the number of movie titles and disks, since for each of these configurations, the only way to provide 500 channels with equal-size clusters is by allocating 2 disks per cluster. The same observation can be made in configurations with more than 21 clusters, which correspond to OMPD (each cluster comprising only 1 disk). Whether clusters comprise one or two disks, disk bandwidth is increasingly under-utilized as the number of cluster grows, since each cluster is used to support less channels, and beyond a certain number of clusters (or disks), the cost constraint will not be met.

In the case of type C constraint, the required movie selection (10 movie titles) is low enough to be supported by most configurations, even by those corresponding to OMPD. The set of eligible configurations is then fully determined by the cost constraint, and corresponds to configurations with $C \in \{1 - 6, 8, 11 - 13, 21 - 27\}$. Constraint A (e.g., 30 channels) is more restrictive since configurations that are acceptable (in the sense of cost) do not necessarily provide enough storage space to store 30 titles. So, both cost and selection constraints have to be considered, making OMPD configurations ineligible. In the limited case of 39 movie titles, only very few configurations are eligible ($C = 1, 3, 4, 5, 6, 8, 13$). It is

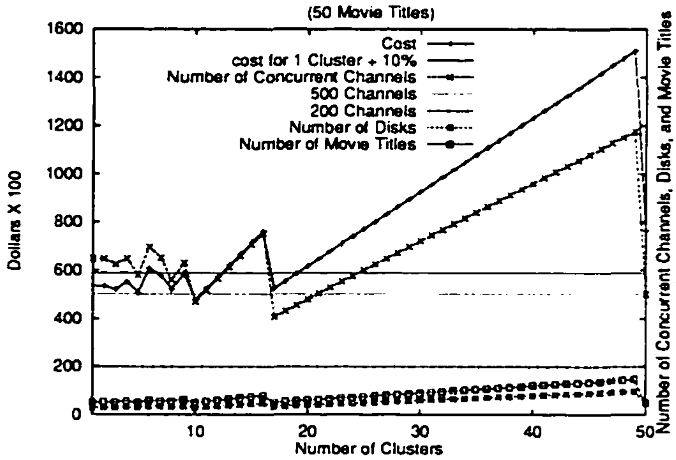


Figure 2.6: Storage cost, number of channels supported and number of disks for various number of clusters: type **B** constraints (50 movie titles required).

worth noting that, even in this restrictive case, the set of eligible configurations would not change significantly if the cost margin is set to 5%. Clearly, a wide range of configurations are possible even when storage design is restricted by a relatively tight cost constraint.

If constraints on channel capacity and movie selection are of type **B** or **D**, the required movie selection can only be supported by configurations whose bandwidth exceeds the channel capacity constraint. As mentioned earlier, such constraints are not typical of front-end VoD servers, and certainly not economically sensible as disk bandwidth may be severely under-utilized. In this case, eligible configurations are found by partitioning the number of movie titles required and keeping the channel capacity of each cluster as low as possible. Similarly to Figure 2.5, we plotted in Figure 2.6 cost, the number of disks, and channel capacity as the number of clusters varies, assuming that a minimum selection of 50 movies has to be provided at the lowest possible channel capacity. Various constraints on the number of channels are also indicated there. Depending on the required channel capacity, there exist several possibilities. First, if a relatively few channels are needed, eligible configurations are determined by cost only. For 200 channels (constraint **B**), eligible configurations are those with $C \in \{1 - 5, 7 - 12, 17 - 19\}$. As the required channel capacity approaches the channel capacity supported by the 1-cluster configuration (590), fewer configurations are eligible ($C = 1, 2, 4, 6, 7$ for 590 channels).

Thus, whenever a service provider decides to increase the capacity of a server, channel capacity should be given priority to viewers' selection so design constraints may be kept in areas **A** and **C** of Figure 2.3. This guideline can be followed by adding RAM first, until full utilization of each cluster, then one disk at a time to each cluster, if necessary.

2.4.2 The Video Allocation Problem

Once an eligible configuration has been chosen, movie titles have to be replicated and assigned to different clusters so that a maximum number of sessions may be supported, and customers' admission latency may be minimized. It is well-known [65] that these goals are achieved in a balanced system by assigning movies to clusters so as to distribute requests uniformly to different clusters. The problem of balancing probabilities for homogeneous accesses across clusters is analogous to the "bin-packing" problem. In the case of equal-size clusters, however, an optimally-balanced system can be obtained once a replication policy has been chosen.

The total storage capacity measured in number of movie titles is compared to the constraint on movie selection. If space is available for replication (e.g., 39 slots for 30 movie titles), unused slots are allocated to videos according to some arbitrary replication policy. A control server will then keep track of the load on each cluster to ensure all the copies to be accessed uniformly. Replication can be done in proportion to popularities, or by simple duplication of the most popular videos. (Note that the latter approach may allow more movie titles to be replicated.) Each video is then assigned a *partial popularity*, which is equal to its initial popularity divided by the number of its copies. Assuming C clusters and s_C slots per cluster, we want to find an assignment from the list of all $C \cdot s_C$ instances of all movie titles and their corresponding partial popularity to the C clusters so the highest access frequency among all cluster is minimized.

We show in Appendix B that optimum video allocation can be found in $O((C^2 s_C^2 + C^3)\log(C \cdot s_C))$ time, starting from any arbitrary assignment. In summary, we show that our assignment problem is reducible to a binary search, in which each iteration involves solving a *max-flow problem* in a directed network [57], thus resulting in a polynomially-bounded complexity. The algorithm presented also works for uneven clusters, provided that cluster capacities are known in advance. In practice, the search space can be greatly reduced by listing all instances of all movie titles (e.g., 38 entries) in decreasing order of partial popularity, then "folding" the sorted list as follows. The first C entries in the list are assigned to the C clusters in ascending order. Then, the next C entries are assigned to the C clusters in descending order. The process is repeated until all entries have been assigned, as illustrated in Figure 2.7. Taking into account the time required for sorting the list of videos ($O(C \cdot s_C \log(C \cdot s_C))$ using Quicksort), the time complexity of optimum video

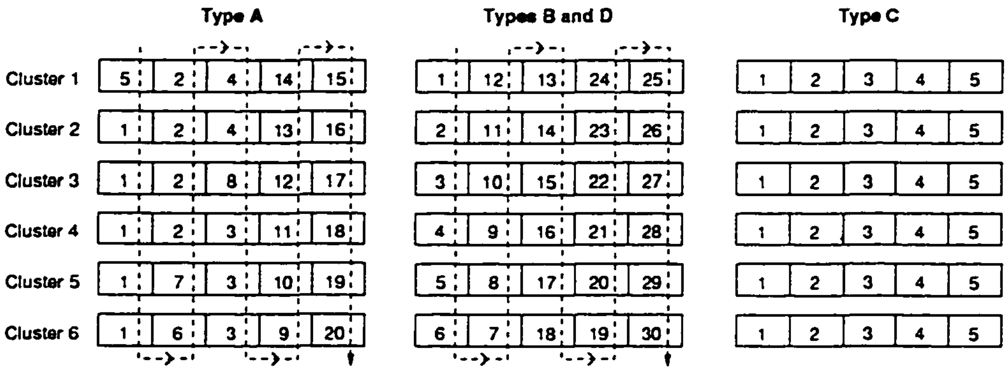


Figure 2.7: Greedy assignment of videos to clusters.

allocation is polynomially bounded.

Depending on the constraint on movie selection, slots may be available for replication and videos duplicated on the same cluster. In such a case, duplexing techniques may be used to provide reliable access in the event of single-disk failures by avoiding duplication of video blocks on the same disk. It should be noted, however, that by avoiding replication of the same video on the same cluster, it is possible to (1) provide reliable access in case of failure of an entire cluster, and (2) accommodate request surges with the completely-shared channel capacity of two clusters. Moreover, duplicating a video in the same striping group does not significantly improve the latency experienced by requests for this particular video in relatively small disk arrays. Since better reliability and scalability are achieved by avoiding replicating the same video on the same cluster, we made sure that no video is duplicated on the same cluster. We also limited the maximum number of replications for a single video to C, as we are interested in the level of reliability provided by replication across clusters.

We now evaluate video allocation by considering the coefficient of variation (or dispersion) in the access frequency of each cluster, as a function of the degree of individuality, or *selectivity*, of videos offered to customers. Selectivity 0 corresponds to a unique movie title, whereas selectivity 1 corresponds to a configurations in which no movie title is replicated. This parameter was chosen in order to illustrate the optimum allocation algorithm in a wide range of replication scenarios. Figure 2.8 depicts the differences in performance between optimum and folding algorithms (FA) in four different configurations, depending on the channel capacity (500 or 1,000) and the number of clusters (5 or 10). An optimum assignment greatly improves over the commonly-used greedy allocation [32, 95] of FA. We also found that only a few steps (typically less than 10) are needed to determine the optimum

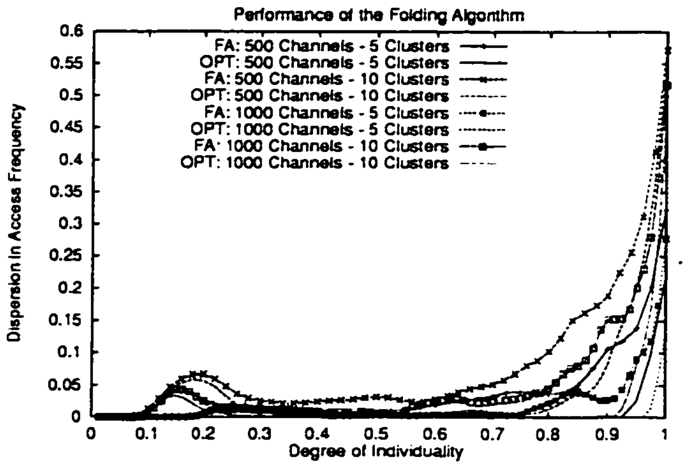


Figure 2.8: Performance of various assignments of videos to clusters.

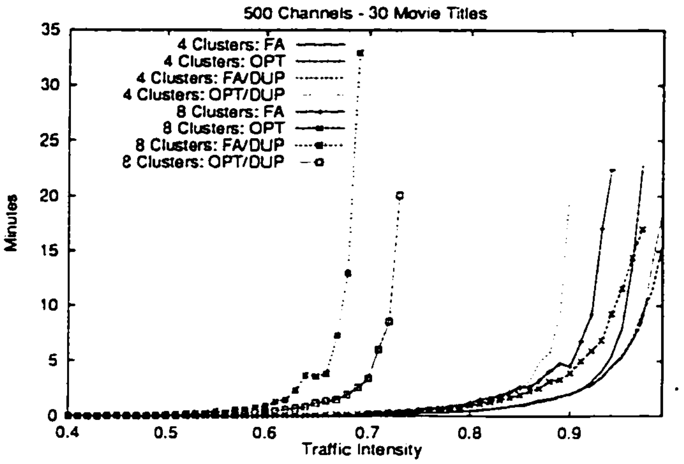


Figure 2.9: Average latency before service: constraint **A**.

video assignment from the assignment initially specified by FA. This is because FA, while being sub-optimal, usually leads to assignments whose dispersion is typically less than 20%. To complete this discussion, in Figure 2.9, we plot simulation results for 4 and 8 clusters (1) FA vs. optimal video assignment, and (2) proportional replication vs. duplication (DUP) with respect to the admission latency in the case of constraint **A**. Clearly, even a slight difference in the dispersion in access frequency between greedy and optimal assignments can result in a noticeable discrepancy in overall admission latency, especially for systems with a large number of clusters. Also, the choice of a replication policy affects the average latency as the number of clusters increases, and in any case, proportional replication is preferable.

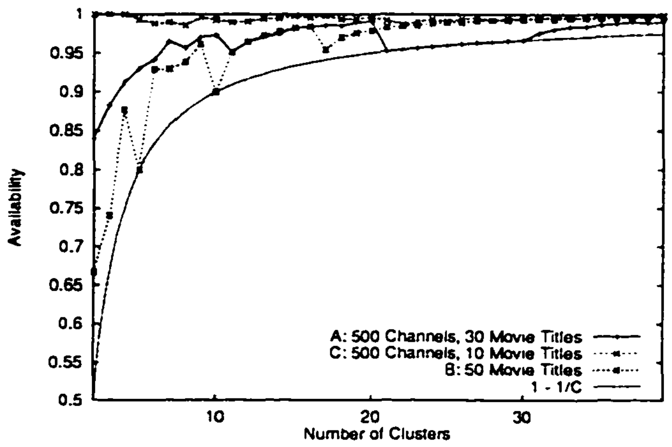


Figure 2.10: Availability for constraints A, B, and C.

2.4.3 Evaluation of Customers' QoS

2.4.3.1 Availability

Service availability is measured by the proportion of customers which remain unaffected by the failure of a single cluster. Assuming that viewers of a replicated video can be shifted to an alternate cluster, availability A_v depends only on the access frequency to non-replicated videos:

$$A_v = 1 - \frac{1}{C} \sum_{non\ replicated} p_i.$$

A_v is plotted in Figure 2.10 for constraints A, B, and C. A_v is lower-bounded by $1 - \frac{1}{C}$, corresponding to the configuration with no replicated movie titles. In constraint **A**, skewed access and moderate replication of the most popular videos in a 2- or 3-cluster configuration is sufficient to ensure $A_v \geq 85\%$. Any additional partitioning beyond 6 clusters ($A_v = 95\%$) improves availability only marginally. In constraint B, variations of A_v are less clearcut since a higher eligible C does not always guarantee greater reliability. In most configurations, however, A_v is strictly greater than the lower bound $1 - \frac{1}{C}$ as the optimum storage capacity is usually achieved at the cost of excess slots, available for replication. In the case of constraint C, a 2-cluster configuration is sufficient to provide near-maximum availability, since most or all of videos can be held within each cluster.

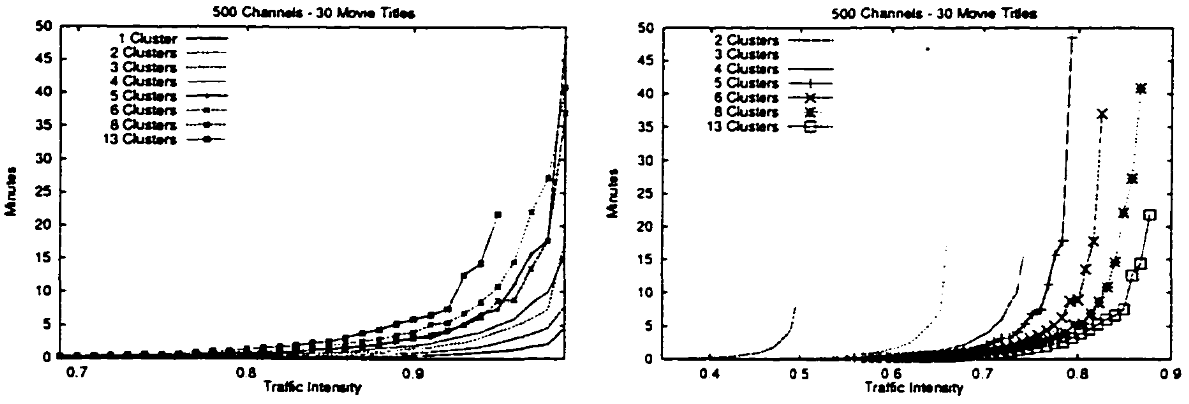


Figure 2.11: Average latency before service and during single-cluster reconfiguration: constraint **A**.

2.4.3.2 Admission Latency

The availability is somewhat misleading since whenever a cluster is affected by a disk failure or reconfiguration, the requests received by the remaining clusters will increase by a factor of up to $1 - \frac{1}{C}$. (This upper-bound may be reached under some constraints of type **C**, depending on the number of clusters.) Replicated videos will thus be accessible with a higher latency, called the *latency during single-cluster reconfigurations* (LSCR), and hence, with very few clusters, the VoD server may only benefit from such high availability for a restricted range of request arrival rate. To investigate this issue further, we plotted in Figure 2.11 , for the various number of clusters, the average admission latency and the LSCR for constraint **A**.

For traffic intensities below 0.8, customers' latency is almost independent of the number of clusters. As the traffic intensity increases beyond this point, the choice of a configuration depends on the desired maximum sustainable traffic intensity, and on the maximum latency and LSCR tolerable by customers. Let's assume, for instance, that the admission latency must remain below 5 minutes. The difference in the maximum traffic intensity guaranteeing such latency between 1-cluster and 13-cluster configurations is approximately 8%. However, the 13-cluster configuration provides near 100% availability and the lowest LSCR among all other configurations. While the 13-cluster configuration represents ȧn extreme case, the choice of a configuration is not as restrictive as it may first appear. In general, as the number of clusters increases, the loss in maximum sustainable traffic intensity is quite tolerable in view of high availability. The choice of a configuration will ultimately depend on which service conditions are expected by the service provider. If reconfigurations are not frequent, choosing a small number of clusters (e.g., 3) provides better scalability since the storage can handle sustained load surges, e.g., during "rush" hour. If, on the other hand, the service provider anticipates frequent reconfigurations (e.g., on a daily basis), several clusters are needed (e.g., $6, 7$ or 8).

In practice, the reconfiguration and request arrival rates, and customers' tolerance to long waits during peak hours, may be difficult to estimate. Thus, the service provider may have to adapt the number of clusters whenever the system is determined to be not operating at the optimal level. The storage can be reconfigured to any eligible configuration if the largest number of disks and amount of RAM required by eligible configurations are allocated. In the case of constraint **A**, the 9-cluster configuration yields the largest number of disks according to Figure 2.5, whereas the configuration with the largest amount of RAM corresponds to one single disk array. Depending on the underlying configuration, excess RAM and disks may then be used to provide a slightly greater channel capacity or movie selection.

2.5 Realistic Traffic Conditions

Thus far, we have discussed stationary request arrival rates; let's consider more realistic time-dependent request arrival rates. For instance, one can observe daily variations in the request rate between "prime time" (e.g., circa 8 p.m.) and "off-hours" (e.g., early morning). On a larger time scale (e.g., one week), changes in movie popularities due to new releases or customers' loss of interest in current titles over time, may also cause changes in the request arrival pattern. Among all other possibly nonstationary factors such as popularity, number of movie titles, or customers' patience — which may not vary in sinusoidal or even in simply periodic fashion — it is safe to assume that time variations will be very moderate compared to that of arrival rate, which varies on a relatively short time scale. Consequently, our general findings in this section should be applicable to a wide range of practical situations.

2.5.1 Choice of a Configuration

The impact of time-dependent request rates for VoD service is usually studied through simulations, as there is, unfortunately, little reliable information available on the workload for VoD systems. As mentioned in [67], many companies are building prototypes to gather such data but are reluctant to share the information because of the competitive nature of the industry, thus forcing storage systems designers to make a "best-guess" estimate

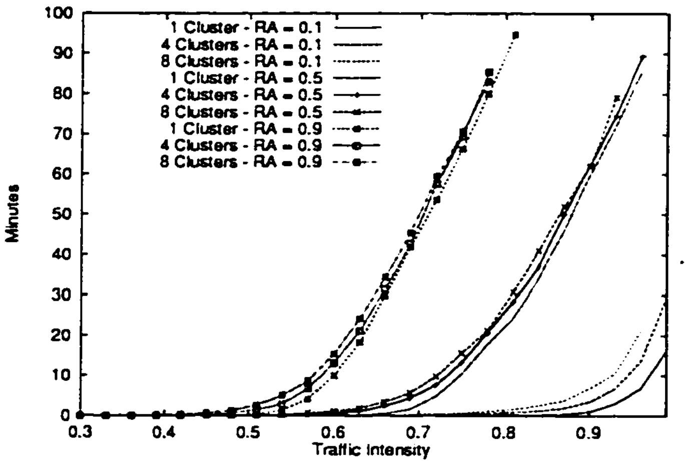


Figure 2.12: Average latency before service for nonstationary arrivals: constraint **A**.

of the workload[5]. For example, normally-distributed requests for a particular movie were assumed in [74], where poor dimensioning is shown to leave a large number of customers unsatisfied during prime time. More general distributions were considered in [11]. We assume for the remainder of this section that the requests aggregated over all movie titles are a time-dependent Poisson arrival process with rate given by

$$\lambda(t) = \overline{\lambda} + A\cos(\frac{2\pi t}{T}). \tag{2.2}$$

where $\overline{\lambda}$ is the daily average arrival rate, $A(>0)$ the amplitude, and T a 24-hour period. So, the rate of requests for movie title m is $\lambda_m(t) = p_m\lambda(t)$. We also note that $\lambda_m = p_m\lambda$ and $A_m = p_m A$. The temporal variations of arrival rate is indicated by the *relative amplitude* $RA = \frac{A}{\lambda}$, which is the same for all movie titles ($RA = 0$ corresponds to the stationary case). As most of the results presented below can be generalized to a wide range of functions, a sinusoidal process is not as limiting as it may first appear. It is sufficient for illustrative and computational purposes and to capture the essence of cyclic customers' behavior.

In Figure 2.12, we plotted the sensitivity of the average admission latency to changes in the request rate for constraint A, in the case of 1, 4, and 8 clusters, and for low, moderate, and high levels of temporal variations (respectively, RA = 0.1, 0.5, and 0.9). For the same value of RA, the maximum sustainable traffic intensity decreases as the number of cluster increases, albeit almost marginally for $RA \geq 0.5$. Our results thus confirm that partitioning is quite advantageous to customers' latency, availability, and LSCR, in case of time-dependent request arrival patterns, since it causes almost no loss in latency. The

[5] Predicting realistic access patterns is further complicated by the fact that these access patterns may be dramatically affected by prices and marketing practices [67].

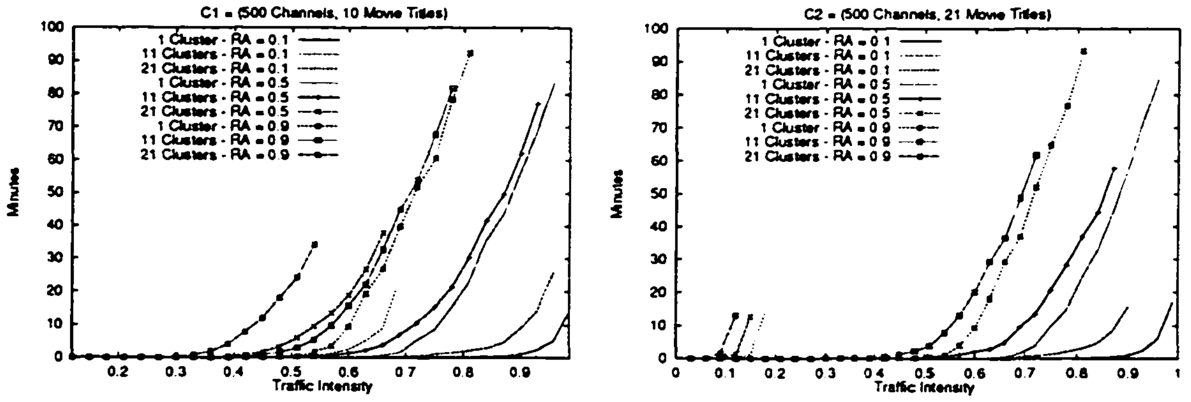


Figure 2.13: Average latency before service for nonstationary arrivals: constraint **C1** and **C2**.

choice of a configuration is also marginally affected by time variations in the request rate.

2.5.2 Motivation for OMPD

As mentioned in Sections 2.3.2 and 2.4.3, OMPD may be economically viable for reliably servicing a large user population with a small number of movies, provided customers' latency is not of paramount importance or the request load remains below the maximum sustainable request rate. The first graph of Figure 2.13 shows that the difference in maximum sustainable traffic intensity for constraint C1 (500 channels and 10 movie titles) between OMPD and 1-cluster configurations is 0.4 for moderately stationary traffic ($RA = 0.1$) and 0.1 for $RA = 0.9$. As illustrated in the second graph of Figure 2.13, however, choosing OMPD in constraint C2 (500 channels and 21 movie titles) results in very low sustainable traffic intensity since no replication is possible. OMPD can thus be advocated with a nonstationary request rate if, as in constraint C1, the number of movie titles is low enough for replication. Should OMPD be used, we show in Appendix C using approximation techniques for queueing systems with nonstationary request arrival rates that near-optimal performance can be achieved with replication of movies in proportion to movie popularities.

2.5.3 Discussion

It is not possible to have exact knowledge of customers' request rate. For completeness, we briefly comment on the role of adaptive techniques to alleviate the congestion caused by random short-term or long-term variations in customers' demand.

In the first case, the dynamic policy of segment replication (DPSR), as proposed in

[32], may be used to balance the load across the clusters. and protect the VoD server from short-term request fluctuations. DPSR partitions video files into a small number of fixed-size segments, stored on various clusters, and dynamically replicated on specially-reserved storage space. It is shown in [32] that for constraints similar to constraint **A**, DPSR can complement static allocation based on duplication and greedy placement of videos across clusters (a 21% improvement in service rate is documented). If, however, relatively few slots are available for replication, such a scheme will not be able to prevent quick saturation of the storage system due to request surges. If, as in constraint **C**, a large number of slots are available for replication, the improvement yielded by dynamic replication over static allocation of slots in proportion to popularities will be marginal. Also, slots available for replication are completely shared by all movie titles. If the request rate exhibits wide or unpredictable temporal variations. saturation can only be prevented through conservative connection admission control as no guarantee can be made on the latency or blocking probability when video segments need to be replicated. Lastly. replication decisions are based on deterministic prediction of future load. i.e.. in a system without out-of-sequence access to videos.

Short-term adaptation techniques should therefore be combined carefully into capacity planning, especially in the case of longer-term variations (e.g.. daily. monthly, or seasonal). Determining whether the system's responsiveness to load surges will be limited by the complexity of long- and short-term adaptation algorithms is still an open issue. It is shown in [95] that predictive video replication. migration. and replication across the storage hierarchy of a distributed video server can efficiently complement adaptive scheduling of requests and dynamic load management. Such techniques are made easier with clustered configurations, since video objects can be reallocated to data sources without noticeable disruption to the existing service.

2.6 Resource Allocation in True-VoD

To be considered as a viable alternative to video rentals. D-VoD servers will have to eventually support VCR functionality. and thus provide the so-called true-VoD (T-VoD) service. T-VoD servers will offer viewers the ultimate flexibility in (1) selecting a video program at any time. and (2) performing any VCR-like user interactions. After admission. customers' interactive behavior consists of the following types of interactions, originally identified in [63]:

Play/Resume: Regular video playout from the beginning or any other location.

Stop/Pause/Abort: Stopping the presentation, without picture or sound (Stop, Abort), or with picture but without sound (Pause). An abort action terminates the connection.

Fast Forward/Rewind: Immediate jump to a particular video location in the forward (backward) direction.

Fast Search/Reverse Search: Quickly moving presentation forward (backward), with picture and possibly sound.

Slow Motion: Moving presentation forward slowly, with picture and, possibly sound.

A T-VoD server may also provide support for other interactions such as *reverse* and *slow reverse*, which correspond to playing a presentation in the reverse direction, at normal or slow speed [63]. In the rest of this dissertation, however, we will not consider these interactions as part of the usual behavior of a customer viewing a video.

Linear access to CBR videos in D-VoD makes it possible to estimate the number of concurrent channels supported by a disk array, then to either accept or reject a storage organization based on cost, the number of movie titles, and channel capacity. In T-VoD, provisions for VCR actions will alter data delivery rate, sequential access, and retrieval scheduling. Serving a customer in interaction or "trick" mode (e.g., during a slow motion, fast or reverse search) requires the VoD server to adopt a specific policy for video retrieval and delivery. Several schemes, outlined below, have been proposed to exploit the characteristics of video streams and, possibly, human perceptual tolerances, in order to support VCR actions with minimum overhead. Considering the general lack of field data in the area of interactive services, researchers have to make assumptions on customer's behavior and tolerance to QoS degradation. Service providers will therefore experiment with VCR functionality through successive upgrades of an initially-specified VoD server configuration. In this section, we present approximation techniques to investigate the expandability of a disk-array-based D-VoD server to T-VoD service, assuming that data layout and disk scheduling remain unchanged; that is, with coarse-grained sequential striping of consecutive stripe units — of optimal size, as determined in [77] — in a round-robin fashion, and C-SCAN disk scheduling.

2.6.1 Probabilistic Connection Admission Control

As mentioned in [26], random seeks caused by VCR actions are easily schedulable in OMPD and FGS, since a (logical-)disk is naturally a random-access medium. In CGS, on the other hand, if the disk array is operating near saturation, non-sequential access or retrieval spanning across several disks complicates scheduling operations since available entries must be in the service list of each target disk during a round. Consequently, data retrieval does not necessarily proceed in lock-step. Here we propose an approximation of the number of concurrent interactive channels supported by a CGS disk array. This approximation can be used for "probabilistic" connection admission control, to protect existing connections from hiccups with a certain confidence level. It is reasonable to assume that disk scheduling and data retrieval in a balanced system are marginally affected when a large number of customers perform pause, stop, fast-forward, rewind, and slow-motion actions, provided these actions are equally likely among customers and throughout the duration of a video. We will therefore focus on fast and reverse search actions, as these actions require the VoD server to adopt a specific retrieval and delivery policy.

Several techniques have been proposed to implement fast and reverse search at n times the normal playback rate. In general, depending on the additional resources required (e.g., file format, storage, network support), the flexibility in the choice of a browsing speed, and in the discontinuity felt by the user, there are three basic types of schemes. First, the simplest scheme retrieves and transmits video frames at n times the normal delivery rate [35], and requires significant additional system resources. The second type skips frames, or a group of frames, if the MPEG standard is used for video storage [25]. In the latter case, the selective segment sampling in [25] allows fast and reverse search in CGS disk arrays transparently, or without significantly interfering with storage throughput, and hence, with the existing connections. However, viewers don't have any flexibility in the speedup factor, and discontinuities are felt during fast scan because the system does not retrieve consecutive blocks. Lastly, based on the assumption that a lowered picture quality is tolerable during fast or reverse search, the third type stores multiple versions of the same movie material corresponding to different picture qualities and rates. This approach is attractive in view of the fact that typical fast-forward scans of VHS video display approximately every sixteenth frame. This type of scheme is made possible by scalable multi-resolution compression algorithms (e.g., MPEG-2) [53], which allow media files to be stored as a base component and a number of enhancement components to support various spatial and temporal resolutions and varying image quality. Non-scalable video compression algorithms such as MPEG-1 can be modified to emulate scalability [88]. An alternative is to append to the original movie a re-compressed copy under forward or backward search playback [70], but at the cost of a significant increase in storage, depending on the choice of a speedup rate.

While it is beyond the scope of this dissertation to determine which scheme is better suited to disk-array-based video servers, it should be noted that for illustrative purposes, we will simply assume that fast or reverse search actions require a retrieval rate of $K_1 R$, where K_1 is the speedup factor. This assumption holds whenever fast and reverse search operations alter the data retrieval rate, for instance, when video frames retrieved at n times the normal delivery rate, a scalable multi-resolution compression algorithm is used, or when separate re-compressed copies are retrieved, because, regardless of the encoding scheme, each stripe unit corresponds to a video block of fixed duration.

Our approximation is based on the assumption that conservative connection admission control through resource over-reservation, or *overbooking*, is needed to accommodate unpredictable fast scans. If the channel capacity of a disk array is calculated based on a display rate $R_c = R + R_o$ slightly greater than normal playback rate R, only a small percentage of connections will suffer from hiccups when the number of customers is large. Let's further assume that the probability of a customer requesting a fast scan is constant throughout the duration of a video and let's consider an arbitrary disk. Denoting by C_d the number of channels of rate R_c supported by an individual disk, the maximum number of fast-scan channels of rate $K_1 R$ is given by $C_t = \frac{R_c}{K_1 R} C_d$. If viewers served by a disk are homogeneous and independent, the number N of customers performing fast or reverse search reduces to a binomial random variable and the *overloading probability* due to fast or reverse search is given by:

$$P(N > C_t) = 1 - \sum_{k=0}^{C_t} \binom{C_d}{k} p_{fs,rs}^{k} (1 - p_{fs,rs})^{C_d - k}, \qquad (2.3)$$

where $p_{fs,rs}$ is the probability of a viewer being in fast or reverse search mode. Clearly, using Eq. (2.3) for connection admission control requires (1) knowledge of customers' behavior to determine $p_{fs/rs}$, and (2) specification of an average channel rate R_c. We address these two issues in the next section by proposing a model of channel usage, which, albeit speculative, combines analytical tractability and potential realism.

2.6.2 Modeling VCR Actions

Various models of customers' interactive behavior have been proposed [35, 63]. While these models differ in the nature of the VCR actions under consideration, they usually

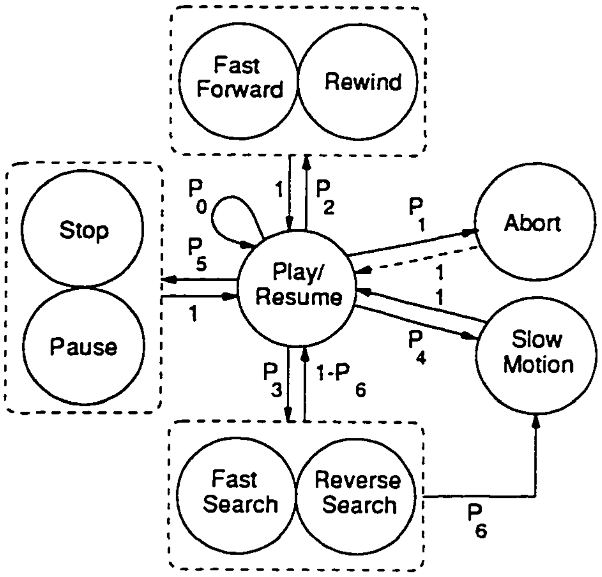


Figure 2.14: Simplified transition diagram for channel usage.

represent customers' behavior by a succession of normal play and interactive states of exponentially- or uniformly- distributed durations. and transitions between interactive states such as slow motion and reverse search are not part of the model. For the sake of realism. a model should capture two specific parameters for their potentially significant impact on the performance of the VoD server and on the channel rate: (1) the level of interactivity. or frequency of requests for VCR actions; (2) the duration of VCR actions.

For this purpose, we assume that from customers' activities one can specify the channel usage model in Figure 2.14. In this model. a set of states corresponding to the different VCR actions are assigned durations and transition probabilities to neighboring states. Fast and reverse search statistics are combined together, as each of these interactions usually involves the same mechanism. The same assumption holds for stop and pause actions, and for fast forward and rewind as well. A channel serves each state for an exponentially-distributed period of time. Since we are interested in the rate of a channel during the service (i.e., once a customer has been allocated a channel), it is safe to assume that there is always a customer ready to take the place of a departing user. Users leave the system either after issuing an abort command or when the end of a movie is reached. We assume that both cases are accounted for in P_1, the probability of a transition from play-mode to abort.

The set of transitions depicted in Figure 2.14 is chosen arbitrarily to show the key aspects of the problem, so that we can glean heuristics that work well in practice. Other sets of states and transitions are also possible, if they turn out to be closer to real behavior.

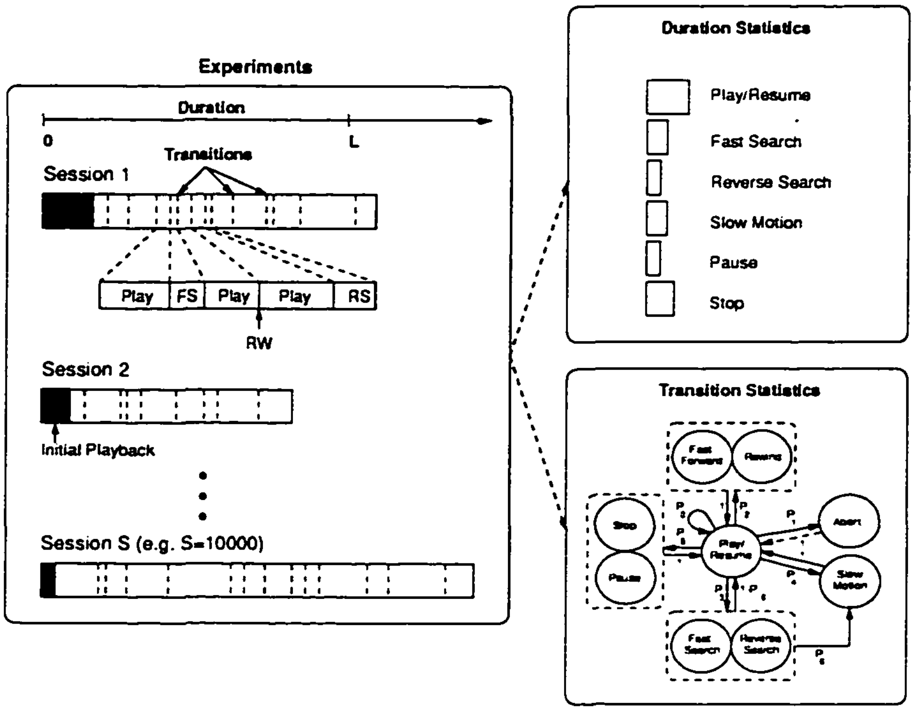


Figure 2.15: Experimental measurements of interactive behavior.

Meanwhile, finding *representative* values for Figure 2.14 is still an open issue since, to our best knowledge, there are no published realistic (or empirically verified as such) models of interactive customers. Until field data collection confirms their pertinence, our assumptions and their proper are therefore inevitable. Also, note that the above-mentioned parameters are captured by this representation of viewers' activity: the level of interactivity can be adjusted by assigning higher or lower transition probabilities from play/resume state to other states and the duration of VCR actions is included in the model. Lastly, transitions between interactive states such as slow motion and reverse search are modeled.

Our channel usage model is applied with the following durations: $d_{InitialPlayback} = \frac{1}{\mu_1}$, $d_{Play/Resume} = \frac{1}{\mu_2}$[6], $d_{Fast/Reverse\ Search} = \frac{1}{\mu_3}$, $d_{Slow\ Motion} = \frac{1}{\mu_4}$, and $d_{Stop/Pause} = \frac{1}{\mu_5}$, and correspondingly, the transition probabilities $P_j, j = 0, \cdots, 6$. These usage statistics can be collected in visualization laboratories, or in libraries with traditional video equipment, by monitoring a particular viewing station for a sufficiently long period and keeping track of the time users spend on a particular operation, counting transitions from one state to another, and computing their relative frequencies. Figure 2.15 summarizes this experimental set-up.

[6] Note that we refined the channel usage model by making a distinction between *initial playback*, which corresponds to customers starting a video, and play/resume actions in the middle of an ongoing presentation, as both types of event could be statistically different.

Behavior	P_0	P_1	P_2	P_3	P_5
NVI	0.65	0.12	0.08	0.08	0.07
VI	0.4	0.14	0.16	0.16	0.14

Table 2.1: Transition prob. from play/resume.

Based on these measurements, we show in Appendix D that the portion of time a particular channel services each interaction, denoted $P(\overline{n}_j), j = 1, \cdots, 5$ is given by $\frac{\frac{\nu_j}{\mu_j}}{\sum_{i=1}^{5} \frac{\nu_i}{\mu_i}}$, where $\nu_2 = \frac{1-P_1}{P_1}\nu_1$, $\nu_3 = \frac{P_3}{P_1}\nu_1$, $\nu_4 = \frac{P_4+P_3P_6}{P_1}\nu_1$, $\nu_5 = \frac{P_5}{P_1}\nu_1$, and lastly, ν_1 satisfying $\sum_{j=1}^{5} \nu_j = 1$. With the knowledge of $p_{fs/rs} = P(\overline{n}_3)$, Eq. (2.3) can now be used for connection admission control.

2.6.3 Evaluation

In our simulation of T-VoD, VCR actions are assumed of equal length and can be either 5-minute long (L) or 1-minute long (S), with the exception of the duration in play/resume state being fixed at 10 minutes and in abort state (of null duration). As represented in Figure 2.14, the transition probability from any state other than play/resume to play/resume is 1, with the exception of $1 - P_6$ set to 0.5. We also assume that slow motion is requested only after a reverse search (i.e., $P_4 = 0$). For other transitions, customers can be either *very interactive* (VI) or *not very interactive* (NVI). Transition probabilities are summarized in Table 2.1. We can now specify four different types of interactive behavior depending on the choice of a level of interactivity (NVI or VI) and duration of VCR actions (S or L).

In Eq. (2.3), specification of an average channel rate R_c can be such that $P(N > C_t)$ remains below an arbitrarily-specified level. Alternatively, we arbitrarily set R_c to the *theoretical channel rate*, which, assuming $\mu_1 = \mu_2$, can be expressed from $P(\overline{n}_j), j = 1, \cdots, 5$ as (noting that stop and pause do not require data transmission):

$$\begin{aligned} R_c &= R(P(\overline{n}_1) + P(\overline{n}_2) + K_1 P(\overline{n}_3) + K_2 P(\overline{n}_4)) \\ &= R\left\{ \frac{\frac{1}{\mu_2} + K_1 \frac{P_3}{\mu_3} + K_2 \frac{P_4+P_3P_6}{\mu_4}}{\frac{1}{\mu_2} + \frac{P_3}{\mu_3} + \frac{P_4+P_3P_6}{\mu_4} + \frac{P_5}{\mu_5}} \right\}. \qquad (2.4) \end{aligned}$$

where $K_1 \geq 1$ is the speedup factor in fast/reverse search and $K_2 < 1$ the slowdown factor. For $K_1 = 3$ and $K_2 = 2$, the increase in channel rate from D-VoD to T-VoD is summarized in Table 2.2. Table 2.3 indicates the corresponding overloading probabilities, which are usually low and therefore acceptable.

Behavior	$NVI-S$	$VI-S$	$NVI-L$	$VI-L$
%	1.3	2.5	6	11

Table 2.2: Increase in channel rate in T-VoD.

Behavior	$NVI-S$	$VI-S$	$NVI-L$	$VI-L$
%	0.01	0.05	0.35	1.25

Table 2.3: Overloading probability due to fast and reverse searches.

2.6.3.1 Comparison T-VoD vs. D-VoD

The first issue to address is how much T-VoD service costs compared to D-VoD. We repeated the analytical approach taken in Sections 2.3 and 2.4 to determine the cost of providing 500 channels in D-VoD and T-VoD with the four types of interactive behaviors mentioned above. Like in Section 2.4, we assumed that cost must be kept below the cost of the 1-cluster D-VoD configuration *plus* 10%. The results, presented in Figure 2.16, show that most configurations that are eligible for D-VoD (i.e., below the cost margin), are also eligible for T-VoD, provided that interactive behavior, or equivalently, channel rate, remains within a certain range (VI-L or R + 11% in our case). Correspondingly, we represent in Figure 2.17 the loss in channel capacity when an eligible 500-channel D-VoD server is used for T-VoD. For each type of interactive behavior, we also indicated the maximum number of channels that can be supported by adding RAM without changing the number of disks of the D-VoD server. The cost discrepancy appears to be mainly in the additional RAM needed to support higher rate T-VoD channels.

The dramatic cost discrepancy between D-VoD and T-VoD in the 11- and 21-cluster configurations in Figure 2.16 shows that for small cluster sizes, allocating the minimum number of disks needed to provide 500 D-VoD channels results in a system that can only be upgraded to T-VoD by adding more disks, because each disk bandwidth is fully utilized. If for instance, we consider the 21-cluster configuration, the only way to support 500 T-VoD channels with 21 clusters is by allocating 2 disks per cluster instead of 1, thus resulting in a severe under-utilization of disk transfer rate. By progressively adding more clusters (e.g., switching from 11 to 12 or from 21 to 22, 23, and 24 clusters), it is possible to accommodate the loss of 10% in channel capacity due to higher-rate T-VoD channels at an acceptable cost.

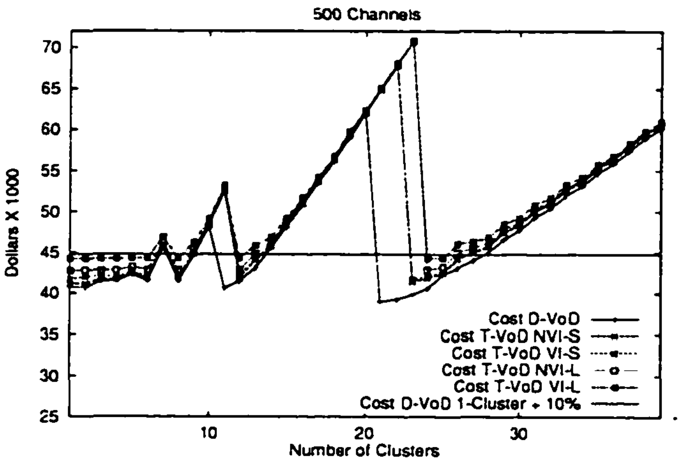


Figure 2.16: T-VoD storage cost for various number of clusters.

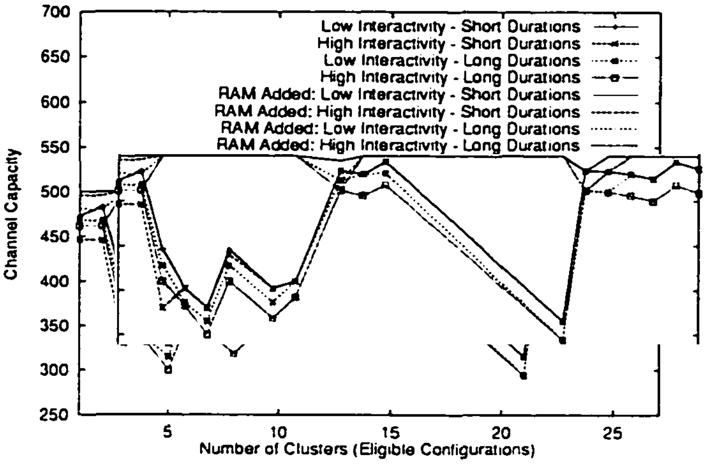


Figure 2.17: Channel capacity of eligible D-VoD configurations used for T-VoD.

Smaller-size clusters can accommodate a number of channels almost proportional to the channel rate. For larger-size clusters with 3 disks or more (i.e., in configurations with less than 10 clusters), the relationship between channel rate and channel capacity is complicated by disk latencies during data retrieval. In this case, Figure 2.17 shows that the loss in channel capacity incurred when a D-VoD server is used for T-VoD is much greater and can drop as low as 300 – 350 channels, even for a 1.3% increase in channel rate in the NVI-S case. Nevertheless, Figure 2.16 shows that the cost of adding RAM, and occasionally disks, to support the same number of T-VoD channels, is acceptable.

2.6.3.2 On Choosing a Storage Organization for T-VoD

Irrespective of the storage configuration used, these results show that it is advantageous to implement fast and reverse search so as not to increase the retrieval rate ($K_1 \leq 1$). This is achievable if fast access is performed by retrieving a version of the movie intended for that purpose [53, 70], or if fast and reverse search is achieved by selectively retrieving groups of pictures [25]. Both solutions implicitly assume that viewers can either tolerate loss in picture quality or discontinuities during fast and reverse search. In the general case ($K_1 > 1$), when a small movie selection is to be supported, choosing between a configuration based on disk arrays and the OMPD scheme depends on how precisely customers' behavior is known. If accurate knowledge of customers' behavior is available, a set of eligible T-VoD configurations can be specified.

If, on the other hand, service providers are experimenting with VCR functionality without prior knowledge of customers' behavior, the VoD server will undergo successive changes of a configuration initially specified for D-VoD service. In this case, reconfigurations in OMPD simply involve adding more disks, and customers' latency, albeit always high, remains relatively constant by switching storage configuration from D-VoD to T-VoD service[7]. In configurations based on disk arrays, switching from D-VoD to T-VoD service requires additional RAM, and sometimes, disks, depending on the cluster utilization. One may then argue that the penalty incurred by re-striping videos on larger clusters can be minimized by storing videos on a larger number of clusters. As illustrated in Figure 2.18, however, even for a reasonably small number of clusters (e.g., 8), using a D-VoD configuration for T-VoD service results in a maximum sustainable request rate comparable to that of OMPD.

2.7 Conclusions

A VoD system must ultimately deliver interactive video services to a large number of customers at a competitive cost relative to broadcast services. This chapter proposed how a distributed VoD server architecture is adapted to become a large-scale service while focusing on long-term resource allocation in front-end servers. Given that VoD service providers target at the home consumer market, we opted for a practical approach and assumed that a large collection of movie titles and concurrent channels should be made available with the

[7] Temporal jumps or fast search throughout a video will change the duration of a connection, which can no longer be considered as deterministic, unless the number of connection is so large that the service time is approximated well to be constant. Nevertheless, simulations indicate that even in local-area front-end servers (e.g., $500 - 1,000$ households), admission latency is primarily affected by the channel capacity and remains almost unaffected by changes in program length due to VCR actions.

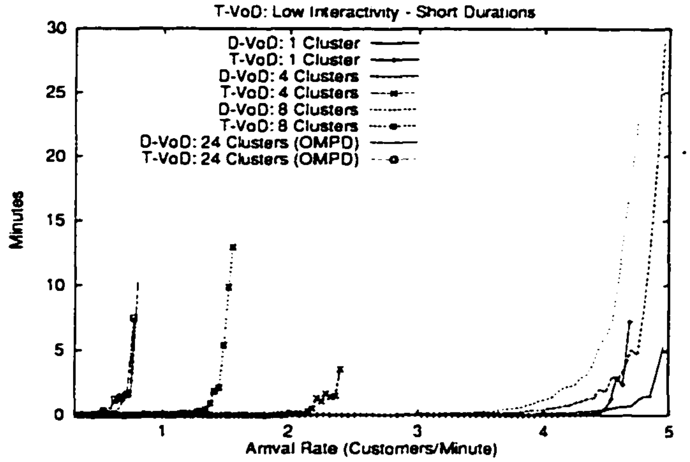


Figure 2.18: Average latency in D-VoD vs. D-VoD used for T-VoD.

highest QoS at the lowest possible cost.

The main contributions are summarized as follows. We first linked customers' QoS to storage cost and capacity by focusing on two basic and well-understood types of storage technologies: OMPD and disk arrays. Both forms of storage constitute two extreme cases and are used to illustrate the range of choices available to local VoD servers. We showed that coarse-grained striping across disk arrays is a potential candidate for its cost-effective storage utilization and low admission latency. Considering that the entire capacity of a disk array is unavailable during reconfiguration, we next investigated hybrid storage organizations, in which videos are partitioned into several clusters. Assuming CGS, we specified a set of eligible clustered configurations and found that, with adequate video allocation across various clusters, a high level of reliability can be provided in a cost-effective way with minor effect on customers' latency. We adapted the specification of eligible clustered configurations to realistic service scenarios, by taking into account temporal variations of the request arrival rate. We were also able to show that the OMPD scheme, often regarded as a naive approach to storage configuration, may actually be quite acceptable for highly reliable service of a small number of movies to a large user population. Finally, we considered T-VoD service, which provides customers with full-fledged interactivity. We proposed a probabilistic connection admission control based on a model of channel usage in T-VoD. Our simulation results indicate that, for a reasonable range of customers' behavior, D-VoD can be upgraded to T-VoD for an increase in cost less than 5%. We also identified practical situations in which approximate knowledge of customers' behavior may justify choosing the OMPD scheme instead of a clustered organization based on disk arrays.

From the VoD server's standpoint, the choice of a clustered configuration in D-VoD or T-VoD, depends on which is valued most, service versatility or scalability. It is, however, possible to provide support for both in a non-conflicting, yet cost-effective way, by batching those requests made by many different viewers for the same movie within a *batching period* [10]. Provided that customers are willing to wait for a duration certainly inferior to that experienced in a saturated system, batching in such multicast VoD systems may potentially increase throughput by serving multiple requests for the same popular movie with a single channel. Furthermore, customers' average latency may also be reduced since requests for the same movie that arrived within the same batching period do not have to compete for resources. Choosing a scalable batching policy, along with a storage organization to support it, and providing full-fledged, yet scalable support for VCR actions in multicast VoD systems, are major research issues that warrant further investigation. These issues are dealt with in the remainder of this dissertation.

CHAPTER 3

SCHEDULING VIDEO PROGRAMS IN NEAR VIDEO-ON-DEMAND SYSTEMS

3.1 Introduction

The need to batch requests for the same movie title together in a VoD system has long been recognized for scalability and immediate deployment [52]. Reduction of per-customer system cost and improvement of system scalability can both be achieved by delaying the VoD server's response to customers' requests made during the same *batching interval* and hence enabling the VoD server to multicast the requested video. In this chapter, we investigate a batching strategy which sources the same material at equally-spaced intervals, called *phase offsets*. This kind of VoD service, in which subscribers who order a particular movie to start within a specific time window are grouped together, is termed "Near-VoD" (NVoD) [12, 65][1]. As a preliminary step in this chapter, we shall restrict our study to monolithic disk arrays, which can devote their entire concurrent channel capacity to any movie title held in storage. The more general case of scalable batching in clustered storage organizations is dealt with in the next chapter.

NVoD is becoming increasingly popular with the telecom, cable, broadcast, set-top box, and computer manufacturers companies (e.g., Digital's *Mediaplex*$^{\text{TM}}$ server [104], Sun's *MediaCenter*$^{\text{TM}}$ *UltraSPARC*$^{\text{TM}}$ server [105], and General Instrument's *MPEG-2 Digital Network System*$^{\text{TM}}$) as it offers the potential to provide scalable, cost-effective digital media services. Because of the regular retrieval pattern, customers can be informed exactly when the transmission will start during the scheduling phase, and experience the same admission latency regardless of the request arrival rate. Thus, even when resources are scarce, NVoD can guarantee *scalable and predictable* response times for all incoming requests. From the

[1] The term "Enhanced Pay-Per-View" (EPPV) is also used by some authors [41, 42, 75].

service provider's perspective, the number of concurrent clients, which may by far exceed the maximum number of concurrent streams, is not upper-bounded by resource availability.

In addition to the improved scalability and cost effectiveness during the scheduling phase, as mentioned in [12], the main advantage of NVoD systems over other batching policies is that, by keeping the batching interval nearly constant per movie title, it is possible to provide D-VoD customers with *limited and scalable VCR capability*. It is usually recognized that full support for continuous interactive functions in a multicast VoD system can only be achieved by dedicating a channel per customer, as in T-VoD, thereby seriously limiting system scalability. In NVoD systems, on the other hand, limited continuity in VCR actions can be provided by caching a small amount of video data (e.g., 5 minutes' worth of video) in the CPE buffer. This buffer can then be accessed without removing the customer from the multicast group. Moreover, staggered phase offsets support for discontinuous VCR actions is provided in NVoD by allowing customers to specify the length of video they want to skip, possibly in integer multiples of the phase offset duration. In this case, the NVoD server will reassign the customer to the multicast group whose playout point is the closest to that requested by the customer. Multicast group membership may also change in a quasi-continuous fashion when customers attempt to access video data outside the CPE buffer as a result of a continuous VCR action; in such a case, the NVoD server will assign customers to an adjacent multicast group. Support for intermittent VCR actions can thus be provided inexpensively in NVoD, and without limiting system scalability. From a service provider's point of view, this feature can be used for preliminary experimentation with interactive service and to investigate custumers' behavior, before making a large investment into T-VoD-like service.

3.1.1 Approach

As we have seen in Chapter 2, the number of concurrent channels supported by a disk-array-based VoD server is usually limited by the underlying storage capacity, organization, and cost. This concurrent channel capacity has to be shared among different movie titles of heterogeneous popularities in the system. We define a *schedule* of video programs in NVoD as the assignment of phase offsets to each movie title. For a concurrent channel capacity of s, a schedule corresponds to a partition $[s_1, \cdots, s_N]$, $\sum_{m=1}^{N} s_m = s$ such that the phase offsets are given by $[\frac{L_1}{s_1}, \cdots, \frac{L_N}{s_N}]$, where L_i is the length of movie i. The number of channels allocated to each movie title depends on its popularity. For instance, in the case of a rarely-requested movie, the NVoD server will probably allocate very few channels to it so that more channels may be allocated to popular titles, and therefore, more customers may be served[2]. Customers' willingness to wait for service will also impose constraints on the maximum acceptable phase offset, and hence on channel allocation. In this chapter, we present heuristics to determine schedules that optimize NVoD server objectives, such as maximum throughput, defined as the average number of customers served per movie transmission, and minimum phase offset, indicating customers' QoS. Owing to their analytical simplicity, these heuristics can easily adapt to changes in workload in a real system, caused by variations in the popularity profile throughout the day, or due to renewal of movie material on a weekly basis.

This chapter is organized as follows. We first provide some background on the feasibility of NVoD and outline how it can be implemented in practice. Next, we analytically derive expressions for the throughput under general conditions of customers' patience and request rates. Based on our analysis, we then present and compare various heuristics to determine optimal schedules. After relaxing the assumption of constant phase offsets, we will show that the throughput of an NVoD server can be improved by using a threshold-based admission control of customers' requests. The cost for such an improvement is that the functionality of discontinuous VCR actions can only be provided *in an average sense*. However, we will show that a dramatic improvement in throughput can be achieved for a reasonably low dispersion of the phase offsets. Lastly, we summarize our contributions in Section 3.7.

3.2 Background on NVoD and Implementation Issues

Current experiments with small scale NVoD systems hint at the technical feasibility of a large-scale deployment in the general framework presented in Section 3.1.1.

3.2.1 Delivery of Multicast Channels

In terms of delivery and reception of video programs, multicast groups in NVoD can be formed by having multiple CPEs listen to the same channel. Assuming a frequency division multiplexed system such as in the already-available CATV, a multicast group is identified by a particular channel, and a customer can join the group by tuning to the appropriate frequency [91]. Delivery can be made possible as presented in [93, 99], by combining high-speed ATM networking technology to CATV support for multimedia services delivery to

[2] This practice can be commonly observed in large video rental stores, which usually have several copies of the most recent releases available on the shelves.

subscribers.

An interesting alternative to sequential multicasting of videos on dedicated channels in NVoD is the permutation-based pyramid broadcasting (PBPB) scheme, presented in [8]. While attractive in terms of admission latency, storage and disk transfer rate requirements, PBPB suffers from several drawbacks that are worth mentioning in order to emphasize the advantages of sequential multicasting in NVoD. In an ordinary pyramid broadcasting scheme, videos are partitioned into contiguous segments of geometrically increasing sizes. The frequencies of transmission for these different segments vary in a manner inversely proportional to their size. Each channel is then assigned a segment size and broadcasts, repetitively, one segment of that particular size from each video. In PBPB, each channel is itself divided into Np subchannnels, where p, number of subchannels allocated to a particular movie title, is determined so as to minimize the required amount of CPE buffer. Each subchannel then sources the same segment at rate $\frac{R}{p}$. One can see that VCR functionality is not easily supported because of the non-sequential transmission of video data and the altered rate within each subchannel. In addition, as documented in [8], hiccup-free display in PBPB may require buffering of about 10% of a movie length in the worst case. Although this is better than pyramid broadcasting, affordability and intellectual property constraints impose limits on the CPE-buffer size. It should be noted, however, that a detailed analysis of PBPB is somewhat tangential to the purposes of this chapter since our primary objective here is to provide a framework for partitioning the concurrent channel capacity among movie titles. In particular, the various heuristics presented hereafter can be used to specify the sizes of the first segment of each movie, and then partition the first logical channel among the N first segments (and then partition the entire concurrent channel capacity by recursion). This issue is left as future work.

3.2.2 Cost-Effective Storage Support for Periodic Retrieval

The various storage organizations presented in Chapter 2 can support periodic retrieval in NVoD. It is, however, a challenging issue to determine a *cost-effective* storage organization in which the amount of disk bandwidth that is effectively scheduled under the given layout and storage constraints is maximized. This is typically the situation facing large-scale NVoD servers that periodically need to re-schedule their offerings to adapt to a changing audience, content, and popularity profile. This optimization problem is greatly complicated by ***heterogeneous*** phase offsets, movie lengths, and, possibly, display rates. As noticed in [41, 42, 75], periodic movie retrievals from storage generate a difficult periodic task scheduling problem that falls within the realm of hard real-time scheduling. As in the previous section, these issues are orthogonal to the purposes of this chapter. whose focus is on the actual specification of NVoD schedules rather than on their cost-effective implementation. Nevertheless, it is important to understand the complexity involved in sourcing videos *exactly* periodically in a cost-effective way, since, as we shall see. there are alternatives to NVoD that perform better while being possibly easier to implement. For completeness we thus briefly outline the work presented in [41, 42. 75].

In order to source the same material at equally-spaced intervals. the NVoD server needs to retrieve video data from a source device with multi-readout capability, so that multiple requests may share the same buffer and read-write head. The schemes presented in [41. 42, 75] partially solve the periodic task scheduling problem by combining (1) cost-effective data allocation, and (2) specific heuristics to determine *NVoD-schedulability*. that is. the feasibility of a schedule. In the first case. *matrix-based* allocation increases the number of clients that can be serviced under NVoD by laying out data based on the knowledge of phase offsets. This representation of video data views each movie as a set of columns. each column being the concatenation of the data retrieved during a scheduling round for each phase offset. and each row corresponding to contiguous portions of video of length a phase offset. By storing each movie in column-major form. and performing retrieval in columns, matrix-based allocation reduces the overhead of disk latency per stream. and therefore buffering and storage bandwidth requirements. In addition to these features. the matrix-based scheme provides a suitable framework for studying cost-effective schedulability of movies with (possibly) different phase offsets. lengths. and display rates under different assumptions about data layout on a storage organization.

It is indeed found that under such data layout, NVoD-schedulability in OMPD (or *clustering* in [41, 42]) and FGS can be formulated as a generalized variant of the 0/1 knapsack, which is NP-hard. Although near-optimal heuristics with low polynomial-time complexity are presented in [41. 42], the applicability of these heuristics is limited. as both OMPD and FGS suffer from severe disk bandwidth and/or storage fragmentation. leading to underutilization of available resources. These problems are partially solved in *Vertical CGS* (VCGS) [28, 75], which reduces the scheduling complexity by declustering each column of the video in matrix form across all disks. so the entire disk array is viewed as a single large disk in a manner similar to FGS. However. increased disk latency overheads render this scheme impractical in large disk arrays [41, 42]. In *Horizontal CGS* (HCGS), on the other hand, a round-robin distribution of data across the disks has the potential of offering a much better scalability and disk utilization than VCGS. This improvement, however, comes at the cost of more sophisticated scheduling problems, which are non-trivial generalizations of the Periodic Maintenance Scheduling Problem (PMSP) [97, 41, 42]. Given that PMSP is known to be NP-complete in the strong sense [16], the authors of [41, 42] present algorithms based on the concept of *scheduling tree structures* to determine all feasible schedules and schedulable subsets of movies. These algorithms can also be used in the dual problem of *capacity planning*, in order to determine, for a given a user profile, the minimum system configuration that can accommodate it.

In summary, HCGS appears to be the most practical combination of storage organization and data layout for scalable and cost-effective NVoD service. It should be noted, though, that there is still no theoretical bound on the worst-case performance of the proposed NVoD-retrieval scheduling heuristics in HCGS, thus complicating cost-effective NVoD scheduling. While unlikely in a wide range of cases, if it has been determined that a schedule is not feasible, a VoD server can delay channels until an available slot opens up, use another (less efficient and cost-effective) storage organization such as OMPD or FGS, or source channels *quasi-regularly* as presented hereafter. Also, the NVoD-schedulability problem may be simplified if movie titles require the same rate and if the number of stripe units per video is a multiple of the number of disks. As mentioned in Chapter 2, this can be enforced by appending typically less than a minute's worth of movie previews or advertisements.

3.2.3 CPE Support for Scalable VCR Actions

In order to understand how limited support for VCR actions is provided in NVoD, one can discriminate two levels of interactivity according to the continuity experienced by customers [12]. *Continuous* interactive functions allow a customer to fully control the duration of interaction, whereas *discontinuous* interactive functions can only be specified for durations that are integer multiples of a predetermined time increment, which, in NVoD, corresponds to a phase offset. Generally speaking, support for continuous interactive functions in NVoD is provided by caching a limited amount of video data close to the user, for instance, in the CPE buffer, so that the user may access it with a very low latency during interaction [12][3]. Discontinuous actions happen in two possible scenarios. First, customers may suddenly exceed the CPE-buffer capacity while performing a continuous action, e.g., rewinding for too long. In that case, the NVoD server will simply transfer the customer to the multicast group corresponding to an adjacent phase offset. Second, customers may

[3] As an example, current STB models made by Scientific-Atlanta comprise a 10 MB buffer [40].

directly specify the length of video they want to skip. in which case the NVoD server will determine the multicast group whose playout point is the closest to that requested by the customer. Note that support for both continuous and discontinuous interactions in NVoD is independent. Continuous actions are taken care of by the CPE buffer, while the general operation of the NVoD server is to assign the customer to another multicast group when needed. This feature is quite advantageous for preliminary experimentation with VCR behavior.

Let's consider the general operation of a CPE buffer during a continuous VCR action. Video frames are received in a synchronous fashion, and those frames already displayed ("past frames") are kept within the buffer, for reverse search and rewind capability. Initially, the playout point will correspond to the most recently-received video frame. Upon pause, stop, fast reverse or rewind within the CPE buffer. the playout point will change. The CPE-buffer manager can then attempt to keep the playout point as close to the middle of the CPE buffer as possible, so there should always be past frames available for reverse search and unplayed frames, available for pause or fast search. This would be a natural choice if "backward" interactions (rewind and reverse search) are as likely as "forward" interactions (fast forward and fast search). If interactive access is dominated by "backward" interactions, the playout point should rather correspond to the most recently-received video frame or group of pictures (GOP). Note that the only mechanism available to the CPE buffer for controlling the playout point is to discard old frames at a rate slower or faster than the arrival rate of frames from the server. Efficient CPE-buffer management should ultimately discard past frames in such a way that a large number of VCR actions can be satisfied without the NVoD server's support. Both customers' interaction latency and load on the NVoD server will then be minimized. The issue of providing efficient CPE-buffer management, which is highly dependent on customers' behavior. is postponed until Chapter 5. in which it is examined in detail.

In general, the size of the CPE buffer will be subject to affordability constraints. As an example, 5 minutes of MPEG-1 compressed video at 1.5 Mbps represent approximately 56.25 MB, which, stored in DRAM for fast access, should cost around $200 in 1997. On the other hand, the phase offset is adjusted by the NVoD server depending on the movie popularities, so as to allocate more channels to the most frequently-requested movie titles. Thus, an important distinction has to be made between movie titles. depending on the relative size of the CPE buffer compared to the phase offset. The NVoD server will probably allocate very few channels (e.g., $2-3$) to a rarely-requested movie. and the phase offset will be much larger than what the CPE buffer can hold. In this case, support for both continuous and discontinuous VCR actions will be extremely limited, and it is safe to assume that group changes will not occur unless expressly requested by customers. For more popular movies, the CPE buffer should be large enough to hold a phase offset's worth of frames so that a group change can possibly be performed in a continuous fashion. Even in this case, the CPE buffer cannot always guarantee a smooth transition between adjacent multicast groups. To illustrate this fact, let's consider a viewer who initiates a fast search shortly after the beginning of a movie. As no future frame will be available in the CPE buffer, the multicast group change will cause the viewer to experience a jump in phase offset. If the CPE buffer attempts to keep the playout point in the middle of the buffer, whereas the play function can be resumed as soon as the the first frame from the new multicast group is fetched, other interactions such as reverse search, pause, stop and slow motion will become fully functional again only once half a phase offset has been fetched. In summary, whether the phase offset is larger or smaller than the CPE-buffer capacity, continuous VCR actions can only be provided intermittently and without any guarantee on the discontinuities experienced by customers (even with proper CPE-buffer management). Thus, *the assumption of a constant phase offset need not hold to guarantee QoS in VCR actions*. We will use this observation in Section 3.6 to show that it is possible to provide the same granularity of discontinuity in VCR actions, as measured by the average phase offset (APO), while increasing the NVoD server throughput by serving more customers.

3.3 Performance Analysis of NVoD

Analytical modeling of batching systems should capture the tradeoffs generally observed in resource sharing systems between customers' and service provider's point of views and conflicting objectives. In this section, we derive the NVoD throughput for a single movie title, defined as the average number of customers served per movie transmission, as a function of the length of a phase offset. As we shall see, the throughput is also a measure of the customers' defection rate.

3.3.1 Stationary Arrivals

We assume that a total of s concurrent channels is available to the NVoD server, and that these s channels have to be partitioned among N movie titles available to the customers. Thus, s_m channels will be allocated to movie m and the corresponding phase offset is $\frac{L}{s_m}$.

where L is the movie length[4]. Popularities can be expressed as a vector of access probabilities $[p_1, p_2, \cdots, p_N], \sum_{m=1}^{N} p_m = 1$, where N is the number of different movie titles [65]. If the total request arrivals are a Poisson process with parameter λ, the request arrivals at movie title m are Poisson with parameter $\lambda_m = p_m\lambda$.

As mentioned in [58], the main problem with admission batching is that partially-patient customers may drop their requests if they wait for too long, thus reducing the server throughput and economical viability. Therefore, in order to evaluate NVoD in realistic situations, we consider partially-patient customers, who may leave the system after waiting a certain period of time. For a VoD service provider, patience behavior is of particular importance since the server throughput will determine the cost per stream of a configuration, and therefore economical viability. In the choice of a patience model, it is safe to assume that VoD customers are not aware of the number of queued customers when they make their requests, nor do they have fixed deadlines. Several patience models adapted to VoD have been proposed in the literature. In [10], for instance, a probabilistic model is assumed, in which customers wait a normally-distributed time (e.g., of mean 5 minutes and standard deviation $\frac{5}{3}$). In [11], customers agree to a deterministic wait threshold before reneging if they wait an additional exponentially-distributed duration (e.g., of mean 2.5 minutes). In [31], a two-parameter model is used to capture the shape of the reneging function, and the maximum amount of time a client will wait before reneging. In practice, however, this model may be harder to match to real data. These examples indicate that the unavailability of field data collection makes it necessary to choose arbitrary (preferably simple) patience models.

Here, we assume that customers time out after a random exponential waiting time. Unlike other patience models, and albeit its apparent simplicity, this behavior was first observed from customers waiting for telephone facilities, and verified for customers waiting for dial tones [60]. Since then, it has been used in several studies dealing with patience modeling [44, 60]. In our patience model, after joining the queue of movie title m, partially-patient customers will agree to wait τ time units or more with the probability $p_w(\tau, \overline{\tau}) = e^{-\frac{\tau}{\overline{\tau}}}$, where $\overline{\tau}$ is the average time customers agree to wait. In general, the patience rate $\alpha = \frac{1}{\overline{\tau}}$ can be assumed independent of the requested movie title. The NVoD service is depicted in Figure 3.1.

A similar system was studied by the authors of [44], under the assumption that the number of requests "accumulated" between two consecutive starts of the service is the

[4] Assumed the same for all movie titles. This assumption only marginally affects the applicability of our heuristics and corresponding results.

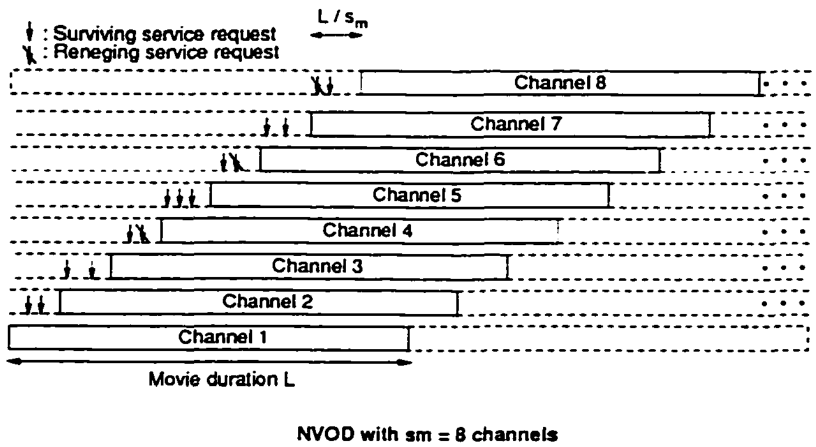


Figure 3.1: NVoD service for $s_m = 8$.

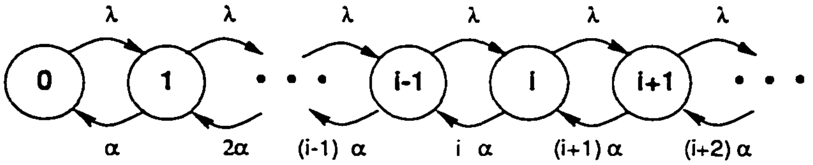


Figure 3.2: Transition diagram for the patience birth-death process.

number of customers who agreed to wait *exactly* $\frac{L}{s_m}$ units of time. In other words, upon arrival, customers are asked if they are willing to wait $\frac{L}{s_m}$, and admitted into the system only if they agreed; had they been admitted, they would have actually waited much less, depending on the time they arrived within the reservation interval before the service phase offset. Our formulation is less restrictive and more realistic than that of [44], in that customers are allowed to make reservations and renege spontaneously. Another way to look at our system is that upon arrival, customers are asked if they are willing to wait for the *exact amount of time τ which separates them from the service interval*, and they accept with a probability $p_w(\tau, \overline{\tau})$.

For the calculation of the number of customers waiting between two consecutive services, we consider the *transient* analysis of the $M/M/\infty$ "self-service" queueing system in Figure 3.2, with arrival rate λ_m and self-service with a negative exponential distribution at rate $\alpha = \frac{1}{\overline{\tau}}$. This system is analyzed in [21], deriving (1) the probability, $P_{M/M/\infty}(t, i, n)$ (Eq. (3.1)), that there are n customers in the $M/M/\infty$ system at time t given there were i customers at time 0 (note that $\binom{i}{k} = 0$ if $k > i$):

$$P_{M/M/\infty}(t, i, n) = e^{-\frac{\lambda_m}{\alpha}(1-e^{-\alpha t})} \cdot \sum_{k=0}^{n} \left\{ \frac{\left(\frac{\lambda_m}{\alpha}\right)^{n-k}}{(n-k)!}(1-e^{-\alpha t})^{n-2k+i} e^{-k\alpha t} \binom{i}{k} \right\}. \quad (3.1)$$

and (2) the mean number in the system at time t (Eq. (3.2)):

$$L_{M/M/\infty}(t, i) = \frac{\lambda_m}{\alpha}(1 - e^{-\alpha t}) + ie^{-\alpha t}. \tag{3.2}$$

The server throughput for movie m per L time units — the average number of service requests granted during L time units — is then obtained by applying Eq. (3.2) to each of s_m phase offsets of movie title m:

$$T_m = s_m \frac{\lambda_m}{\alpha}(1 - e^{-\alpha \frac{L}{s_m}}). \tag{3.3}$$

Let $\rho = \frac{\lambda_m L}{s_m}$ denote the traffic intensity. Then, the throughput variations are linear in ρ for a fixed s_m.

The average loss rate L_{rm} of customers for movie title m due to lack of patience is given by the difference between the average number of arrivals and the number of customers served within $\frac{L}{s_m}$ time units:

$$L_{rm} \;=\; \lambda_m(1 - \frac{1 - e^{-\alpha \frac{L}{s_m}}}{\alpha \frac{L}{s_m}}). \tag{3.4}$$

Note that minimizing the average loss rate maximizes the system throughput. Also, the average defection rate $\frac{L_{rm}}{\lambda_m}$ — the ratio of reneging customers to the total number of customers — depends only on the customers' reneging behavior expressed by α, not on the traffic intensity.

Once admitted, customers move from one multicast group to the next when they request VCR actions that can only be satisfied in a discontinuous fashion. Even though it is possible that no new service requests for a particular time slot were placed in the previous $\frac{L}{s_m}$ time units, requests for that particular time slot may be made later on. Thus, the VoD server has to be *work-conserving*, by restarting service every $\frac{L}{s_m}$ units of time. We want to evaluate the cost of providing work-conserving service, and hence, discontinuous VCR actions capability, or *utilization* of the NVoD server. This can be done by comparing a work-conserving NVoD server with a non-work-conserving one in which, if no new requests for movie m are placed within this interval preceding a particular channel, that channel will be idle for time of duration L. Note that the average throughput is the same in both systems if all parameters are kept the same. The reason for this is that the throughput indicates the number of *new requests* granted every L time units. When no request was made in the previous $\frac{L}{s_m}$ time units, the throughput is unaffected by restarting a channel. Let a *cycle* be the time between two consecutive service starts of a channel, then the cycle in a work-conserving NVoD system is simply the length L of a movie. The cost of providing work-conserving service can be calculated by comparing the cycle lengths of work-conserving and non-work-conserving systems. If they are about the same, then the utilization of the work-conserving NVoD server is very high. On the other hand, a much shorter cycle in the NVoD system supporting discontinuous VCR actions implies that, for the same number of channels, more bandwidth is used to achieve the same throughput as in the non-work-conserving system, thus resulting in a low utilization. An indicator of the extra cost due to low utilization is

$$C_{VCR} = \frac{\sum_{m=1}^{N} s_m \overline{\tau}_m}{sL}, \tag{3.5}$$

where $\overline{\tau}_m$ is the average cycle in the non-work-conserving system.

Now, let's calculate $\overline{\tau}_m$ for the non-work-conserving NVoD system. If we consider one channel for movie m in isolation, the service restarts if at least one request survived during the last $\frac{L}{s_m}$ segment. Hence, with probability $1 - P_{M/M/\infty}(\frac{L}{s_m}, 0, 0)$ the service restarts, and with probability $P_{M/M/\infty}(\frac{L}{s_m}, 0, 0) = exp(-\frac{\lambda_m}{\alpha}(1 - e^{-\alpha\frac{L}{s_m}}))$ the system becomes idle. If $P_{M/M/\infty}(\frac{L}{s_m}, 0, 0)$ is replaced by $P_{0,m}$, we have:

$$\begin{aligned}
\overline{\tau}_m &= (1 - P_{0,m})L + P_{0,m}(1 - P_{0,m})2L + P_{0,m}^2(1 - P_{0,m})3L + \cdots \\
&= (1 - P_{0,m})L\left[\sum_{k=0}^{\infty}(k+1)P_{0,m}^k\right] \\
&= (1 - P_{0,m})L\left[\frac{\partial}{\partial P_{0,m}}\left(\sum_{k=0}^{\infty} P_{0,m}^{k+1}\right)\right] \\
&= (1 - P_{0,m})L\frac{\partial}{\partial P_{0,m}}\left(\frac{P_{0,m}}{1 - P_{0,m}}\right) \\
&= \frac{L}{1 - P_{0,m}} \\
&= \frac{L}{1 - exp(-\frac{\lambda_m}{\alpha}(1 - e^{-\alpha\frac{L}{s_m}}))}.
\end{aligned} \tag{3.6}$$

The cost C_{VCR} in Eq. (3.5) is now fully determined, and C_{VCR} converges to 1 as the arrival rate of requests increases and as customers' patience increases. This observation is consistent with the fact that in both cases, the likelihood of requests' survival from one phase offset to the next increases, thus reducing the number of idle periods.

3.3.2 Nonstationary Arrivals

As we have seen in the previous chapters, in real VoD systems, request arrival rates are usually nonstationary, and alternate between "prime time" and "off-hours" on a daily basis, in addition to seasonal fluctuations on a larger time scale. To account for nonstationary request rates, we must replace the $M/M/\infty$ patience queue in Section 3.3.1 by the $M_t/M/\infty$ system, analyzed in [22, 36].

Fortunately, if we choose an arrival rate function carefully, it is possible to analytically determine the number of surviving customers at the end of a phase offset, which in fact corresponds to *the number of busy servers* of $M_t/M/\infty$ at the end of a phase offset. Similarly to the model used in Section 2.5 of Chapter 2, we assume that the aggregated total request rate for all movie titles (and correspondingly for each movie title m) is sinusoidal:

$$\begin{aligned}\lambda(t) &= \overline{\lambda} + A\sin(\gamma t)\\ \lambda_m(t) &= p_m\overline{\lambda} + p_m A\sin(\gamma t)\\ &= \overline{\lambda}_m + A_m\sin(\gamma t),\end{aligned}$$

where $\overline{\lambda}$ is the daily average arrival rate, $A(>0)$ is the amplitude, $\gamma = \frac{2\pi}{T}$, T being a 24-hour period, and p_m the popularity of movie title m. More general arbitrary models of time variations have been proposed in [11]. Most of these models usually comprise successive intervals with approximately constant request arrival rates over an extended period of time (e.g., 1 hour). To the best of our knowledge, there are no published realistic (or empirically verified as such) models of customers generating nonstationary requests in an NVoD system. Thus, we had to choose an arbitrary model which should, ideally, reproduce a realistic demand on the NVoD server while being computationally tractable. The sinusoidal rate is a convenient and representative model that captures the customers' cyclic behavior; in addition, most of the results presented below can be generalized to a wide range of functions by using decomposition techniques.

As in Section 3.3.1, the main idea in the calculation of NVoD throughput is to model successive phase offsets for one movie title m with an $M_t/M/\infty$ queueing system that gets reset and restarted every $\frac{L}{s_m}$ time units. Eq. (E.3) of Appendix E shows how to calculate the number of surviving requests $L_{M_t/M/\infty}(\frac{L}{s_m}, t_0)$ at the end of a reservation interval of length $\frac{L}{s_m}$ starting at time t_0. In order to express the NVoD throughput, we have to consider all phase offsets during a 24-hour period, or more generally, during a period equal to $LCM(T, \frac{L}{s_m})$ if the number of phase offsets within T is not integer (i.e., a day is not divisible by the movie length). The NVoD throughput for movie title m is therefore:

$$T_m = \frac{1}{\mathcal{T}_m}\sum_{i=0}^{\mathcal{T}_m - 1} L_{M_t/M/\infty}((i+1)\frac{L}{s_m}, i\frac{L}{s_m})s_m. \tag{3.7}$$

where

$$\mathcal{T}_m = \frac{LCM(T, \frac{L}{s_m})}{\frac{L}{s_m}}.$$

Finally, we can express the average loss rate of customers for movie title m by:

$$L_{rm} = \frac{1}{T_m} \cdot \sum_{i=0}^{T_m-1} \left\{ \int_{i\frac{L}{s_m}}^{(i+1)\frac{L}{s_m}} \lambda_m(t)dt - L_{M_t/M/\infty}((i+1)\frac{L}{s_m}, i\frac{L}{s_m}) \right\}. \qquad (3.8)$$

As noticed in [36], if $\lambda(t)$ is a general, not necessarily periodic, function, the analysis of the $M_t/G/\infty$ system can be inferred from the sinusoidal case by combining periodic overlap and Fourier decomposition of $\lambda(t)$. These techniques and their restrictions are elaborated on in [36]. Our formulation of throughput and loss rate can thus be adapted easily to a wide range of nonstationary arrival rate.

Similarly to Section 3.3.1, in order to evaluate the cost of providing discontinuous VCR actions, C_{VCR} given in Eq. (3.5), we need to calculate the average cycle in the non-work-conserving NVoD system. The calculation of the average cycle is simplified by the fact that the number of customers surviving at the end of a phase offset is known to have Poisson distribution. But the average of this distribution varies from epoch to epoch. If we consider a particular phase offset $i < LCM(T, \frac{L}{s_m})$ for movie m, service will restart with probability $1 - P_{0,m,i} = 1 - \exp(-L_{M_t/M/\infty}((i+1)\frac{L}{s_m}, i\frac{L}{s_m}))$. Hence, for this particular phase offset, the cycle specific to that particular phase offset can be calculated by the following recursive formula:

$$\begin{aligned} \tau_{m,i} &= (1 - P_{0,m,i})L + P_{0,m,i}(1 - P_{0,m,i+1})2L + P_{0,i}P_{0,m,i+1}(1 - P_{0,m,i+2})3L + \cdots \\ &= L\sum_{k=i}^{\infty}\left[k(1 - P_{0,m,k})\prod_{l=i}^{k-1} P_{0,m,l}\right]. \qquad (3.9) \end{aligned}$$

with $\prod_{l=i}^{k-1} = 1$ if $k = i$. We found that the product terms beyond the first few terms of the summation are insignificant, and $P_{0,m,i}$'s between adjacent phase offsets are similar. After approximations and calculations similar to those leading to Eq. (3.6), we obtain:

$$\tau_{m,i} \approx \frac{L}{1 - \exp(-L_{M_t/M/\infty}((i+1)\frac{L}{s_m}, i\frac{L}{s_m}))}. \qquad (3.10)$$

The average cycle length can finally be approximated by:

$$\tau_m = \frac{1}{T_m}\sum_{i=0}^{T_m-1} \frac{L}{1 - \exp(-L_{M_t/M/\infty}((i+1)\frac{L}{s_m}, i\frac{L}{s_m}))}. \qquad (3.11)$$

3.4 Optimal Scheduling in NVoD

3.4.1 Problem Statement

We now use the throughput calculations in Section 3.3 to determine partitions of the NVoD channel capacity s so as to maximize the throughput of the server averaged over all movie titles.

Problem NVoD T-OPT: Given s concurrent channels, $\bar{s} = [s_1, \cdots, s_N]$ partition of the channels among the N different movie titles, λ the aggregated request rate, $\mu = \frac{1}{L}$ the constant service rate, $[p_1, \cdots, p_N]$ the movie popularity vector, $\lambda_m = p_m \lambda$ the arrival rate, and $T_m(s_m)$ the throughput corresponding to movie title m, determine

$$\max_{\bar{s}} \sum_{m=1}^{N} T_m(s_m) \quad \text{such that} \quad \sum_{m=1}^{N} \frac{s_m}{s} = 1 \quad .$$

and

$$s_m > 0, \qquad m = 1, \cdots, N.$$

A related problem was presented in [31]. Thanks to a particular patience model, the authors of [31] obtain a simpler expression for the NVoD throughput, which is then used to find an approximate solution to optimal partitioning. Our study offers an interesting alternative to this formalism, since we extend the optimal scheduling problem in NVoD to another patience model for which optimal schedules can be determined *exactly*.

Problem NVoD T-OPT can be further refined. For instance, field studies can determine a range of phase offsets "acceptable" or tolerable to customers, in terms of service latency, discontinuity of VCR actions, and affordability of the CPE buffer. In this case, the objective of the NVoD server is to achieve maximum throughput while satisfying constraints on the maximum and minimum allowable phase offsets for each movie title, in order to provide a service that is "appealing" to the customer. Another objective could be to operate under the lowest C_{VCR} possible. In this case, the objective function can be replaced by $\min_{\bar{s}} C_{VCR}$. Note that constraints on throughput and phase offsets need not be conflicting, since it is possible, in the context of a pricing study, to express the objective function as a weighted sum of both throughput and APO, since both parameters may accounted for in deciding how to charge customers for service for maximum server revenue.

Since the objective function of Problem NVoD T-OPT is convex, and the first constraint is linear, NVoD T-OPT is a discrete convex separable resource allocation problem, and the optimal partition $[s_1, \cdots, s_N]$, denoted by T-OPT, can be found by using integer programming techniques in at most s steps, based on the enumeration technique outlined in Appendix A. This computational efficiency is particularly attractive if the popularity of movies varies over time. In the special case of infinitely-patient customers ($\alpha = 0$), all customers get served within a phase offset. In this case, the NVoD throughput is constant for any partition $\bar{s}$, and one can use another criterion to determine the partition, such as minimizing the APO. In the general case of $\alpha > 0$, minimizing the APO yields a *separate* allocation vector $\bar{s} = [s_1, \cdots, s_N]$ (denoted by EW-OPT), and therefore conflicts with maximization of throughput. (EW-OPT can be determined within s steps similarly to T-OPT). It should be noted that several definitions are possible for the APO. From a random arriving customer standpoint, it is given by $\sum_{m=1}^{N} p_m \frac{L}{s_m}$, which indicates the average waiting time experienced by arriving customers and to some extent the granularity of discontinuity in VCR actions. As seen by a random non-reneging customer (that is, when averaged over all *served* customers), the APO can be defined as $\sum_{m=1}^{N} p_m \frac{T_m}{\sum_{n=1}^{N} T_n} \frac{L}{s_m}$. In this case, it is an indicator of the granularity of discontinuity in VCR actions, and to some extent, of the admission latency. Lastly, from the server point of view (that is, when averaged over all movie titles), it is given by $\frac{1}{N} \sum_{m=1}^{N} \frac{L}{s_m}$. In a sense, this third definition adopts a "fairer", unbiased point of view by not considering movie popularities. In the remainder of this chapter, we shall assume the first definition, in which case the APO indicates the average waiting time experienced by arriving customers. Our methodology and results will remain valid nonetheless if another APO definition is chosen according to whichever point of view is most valued.

The following four allocation policies are considered to evaluate the T-OPT performance while varying the number of channels, traffic intensities and patience.

1. The heuristic NVoD T-OPT which determines the partition *T-OPT* by maximizing the throughput of the NVoD server.

2. A proportional allocation policy, which assigns each movie the number of channels in proportion to its popularity determining a partition *T-PROP*.

3. The heuristic which allocates the number of channels proportional to the square root of popularity, because it was found in [10] that batching customers' requests according to the maximum factored queue length (MFQL)[5] leads to minimal customers' latency in case of infinitely-patient customers. The corresponding partition is denoted by *T-SQRT*. In case of discrete pre-determined phase offsets, even though T-SQRT may not be optimal, it might offer an interesting tradeoff between throughput and APO.

4. The allocation policy which minimizes the APO. The resulting partition is denoted by *EW-OPT*.

[5] For a particular movie title, the MFQL is the queue length divided by the square root of the title popularity

3.4.2 Simulation Results

We chose to compare the throughput (given by Eq. (3.3)), the APO $\overline{EW}$, and the dispersion D — defined as the coefficient of variation of the phase offsets — of the various above-mentioned policies:

$$D = \frac{\sqrt{\sum_{m=1}^{N} p_m \left(\frac{L}{s_m}\right)^2 - \overline{EW}^2}}{\overline{EW}}. \tag{3.12}$$

with $\overline{EW} = \sum_{m=1}^{N} p_m \frac{L}{s_m}$. Lower values of the dispersion D indicate more homogeneous allocation policies which lessen variations in the customers' waiting times. We adopt the Zipf's law as the stationary model of movie popularity.

The throughput given by Eq. (3.3) in the stationary case, and that by Eq. (3.7) in the nonstationary case, are quasi-linear in traffic intensity. An important consequence of this property, confirmed by our simulations, is that once an optimum allocation $\overline{s}$ for T-OPT or EW-OPT has been determined for a given traffic intensity, *it will not change for any other value of traffic intensity*, under the condition that the total number of channels s and the patience factor, defined as $\beta = \frac{\overline{T}}{L}$, are kept constant. By combining both throughput linearity and allocation invariance with traffic intensity, one can now observe that the ratio of throughputs of any two partitions chosen among T-OPT, EW-OPT, T-PROP and T-SQRT is independent of traffic intensity. Consequently, in order to evaluate different allocation policies for a given channel capacity s and patience factor β, it is enough to simply assume an arbitrary traffic intensity. Then, different partitions can be compared with respect to their *relative* throughputs, *normalized with the throughput corresponding to an arbitrary partition*. For this effect, we selected EW-OPT to be the normalizing partition, since it corresponds to the lowest throughput, as we shall see. Note that the allocation invariance property also implies that the APOs and the corresponding dispersions of each partition are independent of traffic intensity.

We measured the variations of the normalized throughput, the APO and the corresponding dispersion in various cases of channel capacities shared among 10 movie titles. Two values of the patience factor are considered: (1) $\beta = 0.01$ corresponding to customers who are willing to wait 1 minute on average, thus *very impatient*; (2) $\beta = 0.1$ corresponding to moderately patient customers, willing to wait 10 minutes on average. (Note that this definition of customers' behavior is arbitrary and used only for an illustrative purpose.)

T-OPT provides an upper bound on the maximum achievable throughput, and EW-OPT a lower bound on the minimum APO, and the corresponding dispersion. Intuitively,

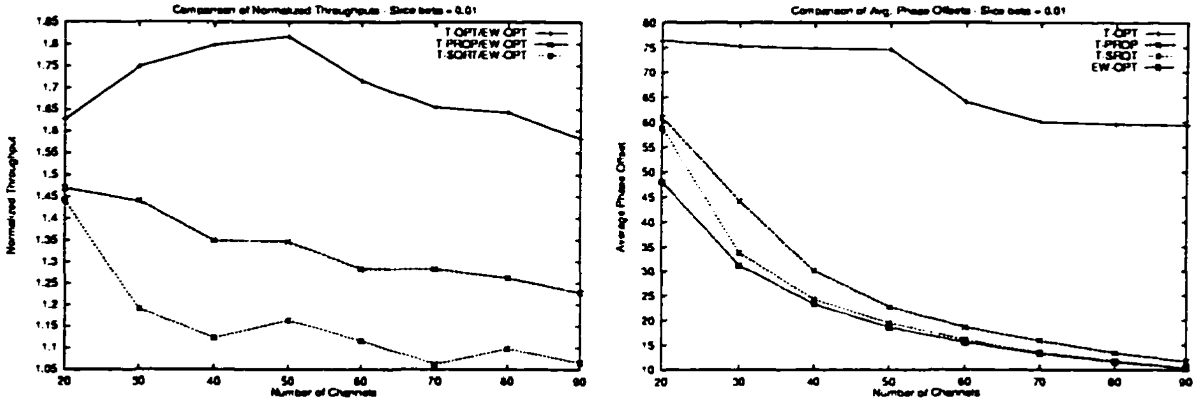


Figure 3.3: Normalized throughput and APO for $\beta = 0.01$.

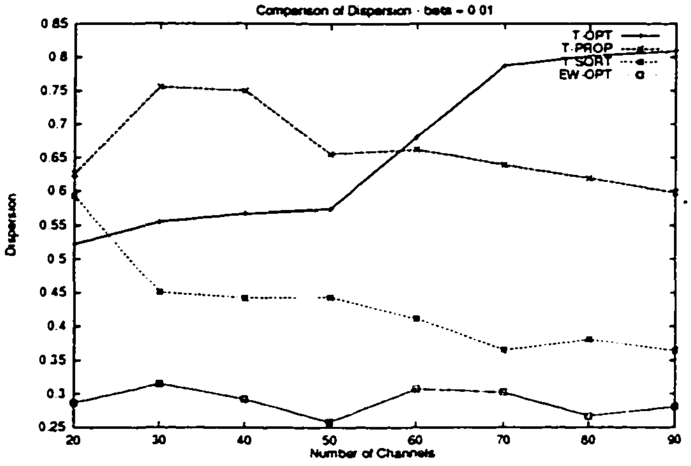


Figure 3.4: Dispersion for $\beta = 0.01$.

for very impatient customers, T-OPT will tend to allocate all channels to the most popular movie title, thereby increasing dispersion and APO. Figures 3.3 and 3.4 clearly demonstrate this pronounced tradeoff between throughput and dispersion for very impatient customers. In such situations, T-PROP appears to be a good compromise.

For more patient customers ($\beta = 0.1$), Figures 3.5 and 3.6 indicate that the throughput is less sensitive to the choice of an allocation policy, as the distinction between policies is less clearcut. T-SQRT then represents a good tradeoff among throughput, APO and dispersion. To summarize our simulation results, the choice of a allocation heuristic by the NVoD server will depend on the number of channels available, customers' behavior expressed by β, and finally, on performance parameters such as throughput, APO, dispersion, or a tradeoff among all of them. We found that the latter case can be achieved with simple heuristics such as T-PROP for impatient customers, and T-SQRT for moderately to very

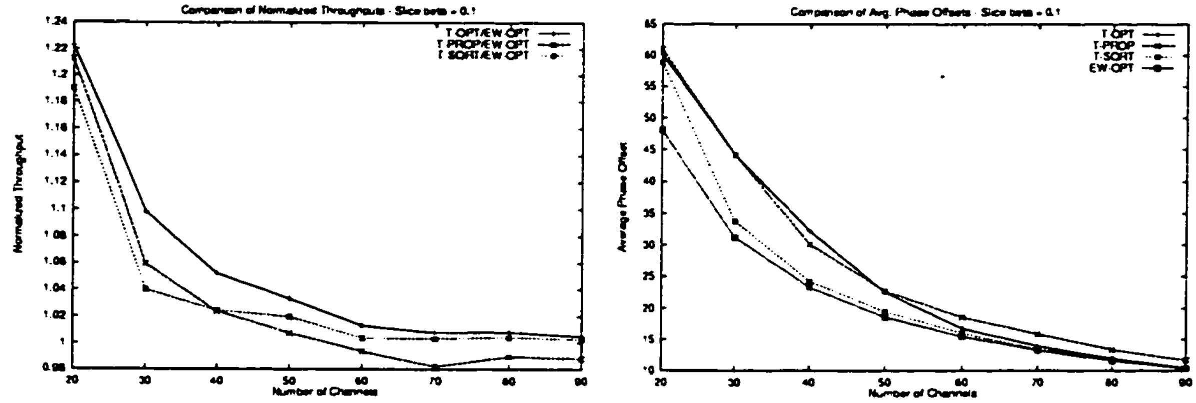


Figure 3.5: Normalized throughput and APO for $\beta = 0.1$.

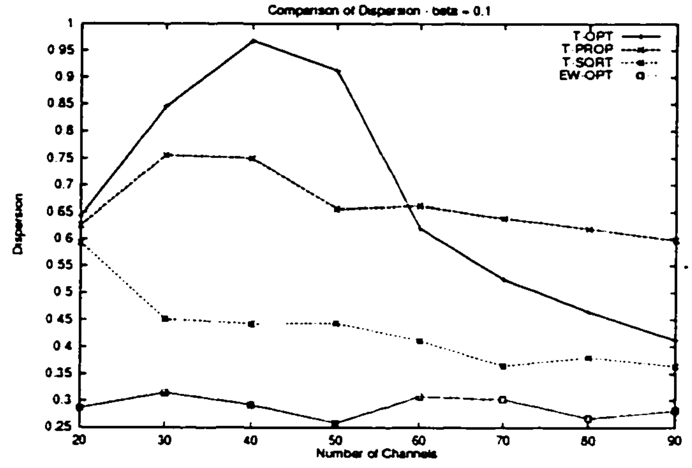


Figure 3.6: Dispersion for $\beta = 0.1$.

patient customers.

Finally, we compare T-OPT, T-PROP, T-SQRT and EW-OPT with respect to C_{VCR} given in Eq. (3.5), which is an indicator of the additional bandwidth needed to provide work-conserving service and discontinuous VCR actions support. A low C_{VCR} value represents allocations for which work-conserving scheduling of channels is less costly than non-work-conserving scheduling. Figure 3.7 shows the simulation results for a request arrival rate $\lambda = 5.0$. (Consistent results were obtained in other simulations with different arrival rates.) For very impatient customers ($\beta = 0.01$), T-OPT exhibits the highest system utilization, since it assigns most of the channel capacity to the most popular movie, which accounts for most of C_{VCR}. The difference between work-conserving and non-work-conserving cycle length is then reduced. For similar reasons, EW-OPT performs worst as it tends to assign channels uniformly. As we kept increasing the patience factor ($\beta \geq 0.01$), T-PROP yielded

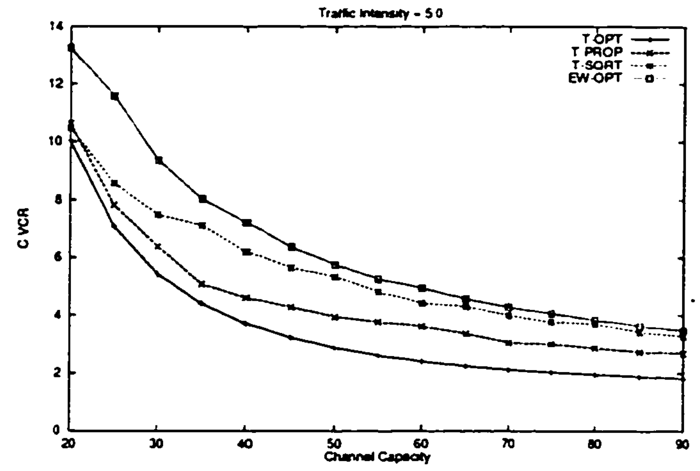


Figure 3.7: Comparison of C_{VCR} for $\beta = 0.01$.

the best utilization, followed by T-SQRT. This serendipitous result indicates that it is a sensible decision to choose the heuristics that provide a good tradeoff among throughput, APO and dispersion.

3.5 Threshold-Based NVoD

We present in this section an intuitively-appealing alternative to fixed-length phase offsets. Suppose service is provided only when no less than K_m requests are "accumulated" for movie title m and a channel is available. If a request is placed while all channels are busy, the customer has to wait for a channel to become available, and also until the desired number of requests is accumulated. If, by the end of the movie length L, there are at least K_m customers waiting then the service restarts; else the service is delayed until the number of requests reaches exactly K_m. We call such class of VoD systems *threshold-based NVoD*, or *Quasi-VoD* (QVoD). We want to compare the performance of QVoD to that of NVoD.

3.5.1 Performance Analysis

Similarly to the approach taken in Section 3.3, our first analytical step is to study the throughput of a QVoD server by focusing on the channels allocated to one particular movie title. For tractability, we will restrict our analysis to the case where a single stream is allocated to a particular movie title m, i.e., $s_m = 1$, and then generalize the results to any number of servers.

Assuming partially-patient customers and stationary arrivals, at the end of the service interval L, the service resumes with probability $1 - \sum_{i=0}^{K_m-1} P_{M/M/\infty}(L,0,i)$, where $P_{M/M/\infty}(L,0,i)$ is given by Eq. (3.1). With probability $\sum_{i=0}^{K_m-1} P_{M/M/\infty}(L,0,i)$ the system becomes idle, waiting until the number of waiting customers reaches K_m. As noticed in [44], if the number of surviving customers at the end of the service interval is k, the length of the idle interval will be the *first-passage time* of the patience birth-death process in Figure 3.2 from state k to state K_m:

$$t_{k,K_m} = \sum_{i=k}^{K_m-1} t_{i,i+1},$$

where t_{N_1,N_2} is the mean first-passage time from state N_1 to state N_2, given by the following recursive formula [44, 49]:

$$\begin{aligned} t_{i,i+1} &= \frac{1 + i\alpha t_{i-1,i}}{\lambda_m} \\ &= \frac{\sum_{k=0}^{i} \frac{(\frac{\lambda_m}{\alpha})^k}{k!}}{\lambda_m \frac{(\frac{\lambda_m}{\alpha})^i}{i!}}. \end{aligned}$$

The mean idle time can now be computed as:

$$\overline{\tau}_{im} = \sum_{k=0}^{K_m-1} t_{k,K_m} P_{M/M/\infty}(L,0,k).$$

Having expressed the mean idle time, the average cycle duration is $\overline{\tau}_{cm} = \overline{\tau}_{im} + L$ and the average number of requests served per cycle is (with $\mathcal{P}_k = P_{M/M/\infty}(L,0,k)$ and $\mathcal{L} = L_{M/M/\infty}(L,0) = \frac{\lambda_m}{\alpha}(1 - e^{-\alpha L})$):

$$\begin{aligned} N_{cm} &= (1 - \sum_{k=0}^{K_m-1} \mathcal{P}_k) \sum_{k=K_m}^{\infty} k\mathcal{P}_k + K_m \sum_{k=0}^{K_m-1} \mathcal{P}_k \\ &= (\mathcal{L} - \sum_{k=0}^{K_m-1} k\mathcal{P}_k) + (K_m - \mathcal{L} + \sum_{k=0}^{K_m-1} k\mathcal{P}_k) \sum_{k=0}^{K_m-1} \mathcal{P}_k. \end{aligned}$$

Finally, the throughput for movie title m, defined as the ratio of N_{cm} to $\overline{\tau}_{cm}$, can be expressed as:

$$T_m = \frac{L}{L + \sum_{k=0}^{K_m-1} t_{k,K_m} \mathcal{P}_k} \cdot \left\{ K_m \sum_{k=0}^{K_m-1} \mathcal{P}_k + (\mathcal{L} - \sum_{k=0}^{K_m-1} k\mathcal{P}_k)(1 - \sum_{k=0}^{K_m-1} \mathcal{P}_k) \right\}. \quad (3.13)$$

The average loss rate, due to impatient customers who leave the queue without receiving service, is simply given by $\lambda_m - \frac{N_{cm}}{\overline{\tau}_{cm}}$. Determining an analytical closed form of T_m and other performance measures such as the average latency is usually intractable for $s_m > 1$, especially in the case of deterministic service[6]. In addition, our simulations indicate that

[6] For exponentially-distributed service time and infinitely-patient customers, an analytical closed form of average latency as a function of K_i is presented in [89] based on the work of Neuts and Nadarajan [69].

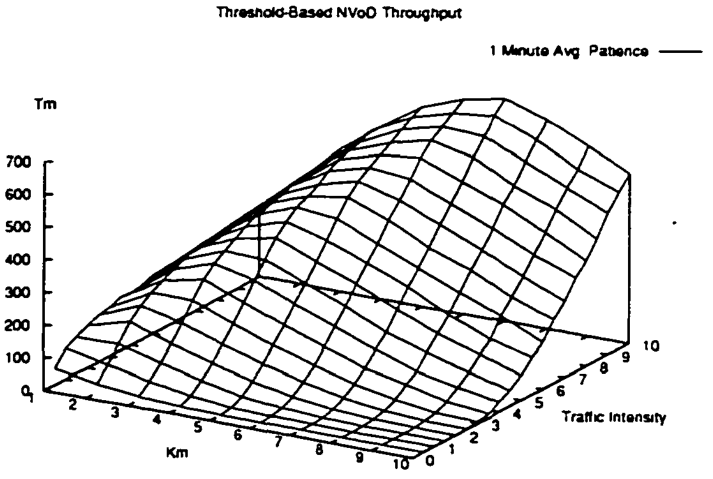


Figure 3.8: QVoD throughput for 100 channels and $\beta = 0.01$.

approximations based on Eq. 3.13 are not satisfying in a wide range of channel capacity and patience factor. Although some guidance may be found in the time-consuming algorithmic procedures presented in [85, 23, 66, 86], simulations or heuristics are needed, based on customers' request rate and behavior.

Nevertheless, some interesting preliminary results can be obtained in the general case $s_m \geq 1$. For a given traffic intensity, small threshold values K_m lead to a sub-optimal throughput due to under-collected service requests, while large values of K_m may cause losses of customers due to long waits. Consequently, there is an optimal value of the threshold K_m^{opt} which maximizes the QVoD server throughput for movie title m. Figure 3.8 illustrates this phenomenon for 100 channels allocated to movie title m. Our simulation results also indicate that K_m^{opt} plays a critical role in achieving the lowest request defection rate. This observation is particularly important if the traffic intensity and customers' patience vary with time (e.g., in a 24-hour cycle). As can be seen in Figure 3.8 and confirmed in separate simulations, K_m^{opt} is linear in traffic intensity, with a coefficient of variation which depends on the number of channels and the patience factor[7]. It is therefore possible to tabulate K_m^{opt} and dynamically adjust K to K_{opt} corresponding to the instantaneous arrival rate of requests. In the next section we evaluate the effectiveness of such an adaptive approach in a nonstationary environment.

[7] This is illustrated in more detail in the next chapter.

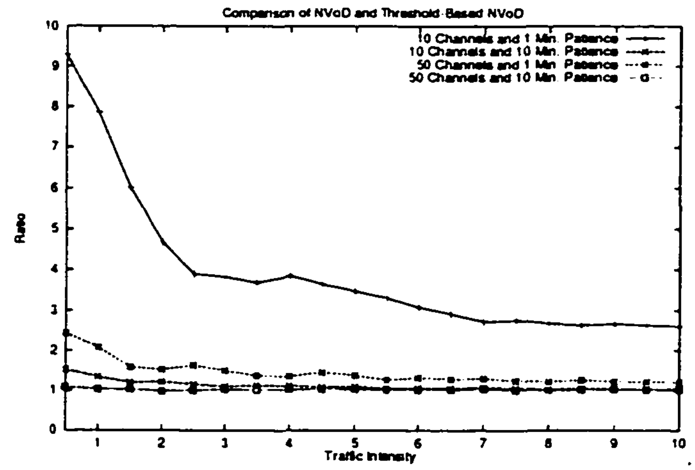


Figure 3.9: Throughput ratio for stationary arrival rates.

3.5.2 Throughput Comparison QVoD vs. NVoD

Relaxing the constant-phase-offset constraint in NVoD by using QVoD is justifiable if the resulting throughput gain is significant. Thus, we now compare the throughput for one movie title in both NVoD and QVoD systems. A complete comparison requires evaluation of both stationary and nonstationary traffic intensities. In the stationary case, Figure 3.9 represents the ratio of the QVoD throughput to that of NVoD, for four different combinations of the channel capacity s_m allocated to movie m, and of the patience factor β. For each traffic-intensity value, the QVoD throughput corresponds to the optimal threshold K_m^{opt}. These results indicate a higher throughput in QVoD systems, although the difference between NVoD and QVoD diminishes as both customers' patience and channel capacity increase. This trend can be explained by the fact that, when the average customers' patience is comparable to, or greater than, half an NVoD phase offset (e.g., 1 minute for 50 channels), the NVoD throughput will be pretty high, hence lessening any relative improvement from using QVoD.

QVoD appears superior to NVoD if the optimum K_m^{opt} is used. However, our simulation results indicate a sharp decrease in QVoD performance when non-optimal values of K_m are used for a particular traffic intensity. This raises questions regarding the applicability of QVoD in case of nonstationary request arrival rates. We also noticed that the defection rate in QVoD is very sensitive to variations of traffic intensity, the patience factor and K_m. In the NVoD system, on the other hand, defection rates depend only on the customers' patience.

There are two policies that a QVoD server can adopt in a nonstationary environment.

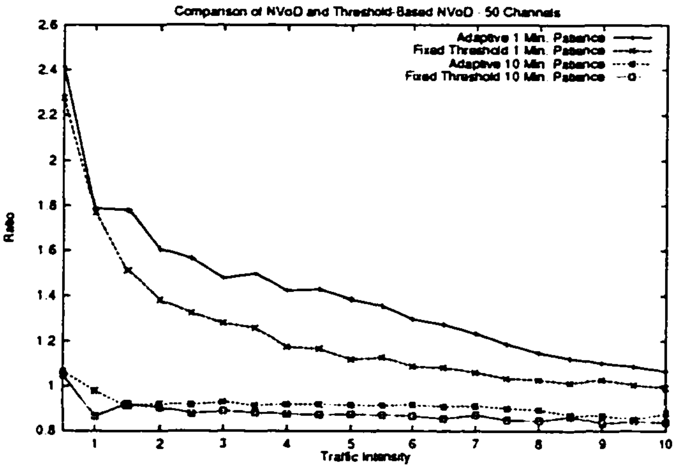


Figure 3.10: Throughput ratio for nonstationary arrival rates, 50 channels and $RA = 0.9$.

First, as mentioned in the previous section, the threshold K_m can be dynamically adapted by choosing K_m^{opt} for the instantaneous arrival rate. Alternatively, the QVoD server can choose the fixed threshold which maximizes the throughput averaged over a 24-hour period. We used simulations to evaluate the ratio of the QVoD throughput in each approach to that of NVoD, for different values of the patience factor. We assumed sinusoidal arrival rates of relative amplitude $RA = 0.9$, which represent an extreme case of time variations. The NVoD throughput was calculated from Eq. (3.7), whereas the QVoD throughput was obtained through recursive simulations. The simulation results in Figure 3.10 for $s_m = 50$ show that, as the number of channels and the patience factor increase, QVoD becomes less attractive for both choices of the threshold, adaptive or fixed.

This conclusion confirms that NVoD should not, in general, be dismissed in favor of QVoD for nonstationary arrival rates. The similarity of performance between adaptive and fixed threshold QVoD policies indicates that threshold-based scheduling of videos is not well adapted to continuously-changing load conditions. Also, unless a closed-form equation is found for key QVoD performance variables such as throughput or average latency as functions of K_m and s_m, recursive simulations must be used to tabulate K_m^{opt} as a function of s_m and β. This process complicates heuristics for determining optimal schedules. Finally, there is no guarantee that a schedule will remain optimal in spite of variations in the request arrival pattern, and changes in the popularity profile thus require new tabulations.

3.6 "QVoD-Enhanced" NVoD

As seen in Section 3.2, even with proper CPE-buffer management and constant phase offsets, continuity in VCR actions can only be provided intermittently and without any guarantee on the discontinuities experienced by customers. Thus, it is intuitively appealing to relax the assumption of constant phase offsets in NVoD for a higher throughput, as long as the same average granularity of discontinuity in VCR actions, as measured by the APO, is provided to customers. We show in this section that by using QVoD over a partition $[s_1, \cdots, s_N]$ of the capacity s among the N movie titles *initially determined for NVoD*, one can achieve a much higher throughput while providing support for discontinuous VCR actions comparable to that of NVoD in an average sense. We call such system *"QVoD-enhanced" NVoD*. We will restrict our analysis to stationary request rates, although it is valid in most nonstationary cases.

Suppose the NVoD server will initially determine a partition $[s_1, \cdots, s_N]$ by optimizing an arbitrary pre-determined objective. This objective may be to maximize throughput, minimize the APO, or to make a tradeoff between throughput and APO. We examine how NVoD performance will be affected by switching to QVoD based on the same $[s_1, \cdots, s_N]$, and the corresponding vector of optimal thresholds $\overline{K}^{opt} = [K_1^{opt}, \cdots, K_N^{opt}]$. According to the results obtained thus far, we can make performance improvement for moderately stationary arrival rates, impatient customers and a relatively small number of channels allocated to each movie title. In a sense, this approach combines the best of both batching systems, partitioning simplicity in NVoD, and throughput performance in QVoD.

We approach the problem by using QVoD in conjunction with NVoD in three experimental steps: (1) First, we have to select arbitrary NVoD partitions $[s_1, \cdots, s_N]$. Since we are interested in customers' QoS, we choose NVoD EW-OPT for minimization of the phase offset, and NVoD PROP or NVoD SQRT for the tradeoff between throughput and phase offset, depending on the value of the patience factor. These partitions were presented in Section 3.4.1; (2) Next, we evaluate performance by switching from NVoD to QVoD; (3) Finally, we compare the performance of QVoD with that of NVoD whose partition $[s_1, \cdots, s_N]$ corresponds to NVoD T-OPT, presented in Section 3.4.1. NVoD T-OPT is used as an indicator of the maximum throughput achievable with a NVoD server.

In summary, we compared the following five systems.

1. An NVoD server with $[s_1, \cdots, s_N]$ minimizing the average NVoD phase offset; this configuration is called *NVoD EW-OPT*.

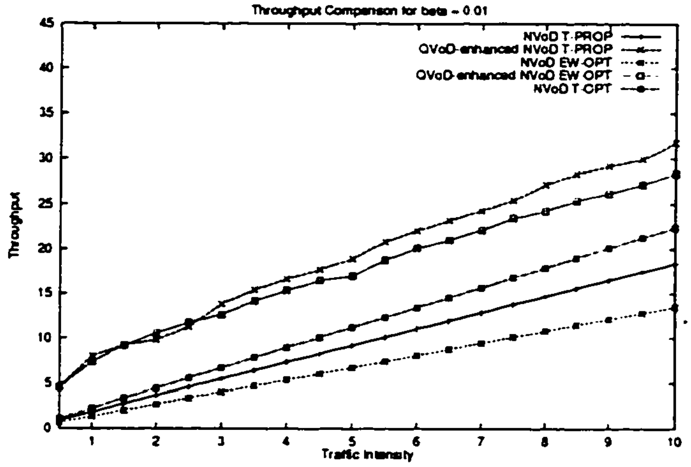


Figure 3.11: Comparison of NVoD and QVoD-enhanced NVoD: throughput for $\beta = 0.01$.

2. A QVoD server with the same channel allocation vector $[s_1, \cdots, s_N]$ defined by NVoD EW-OPT, used in conjunction with $\overline{K}^{opt}$: this configuration is called *QVoD-enhanced NVoD EW-OPT*.

3. An NVoD server with the partition $[s_1, \cdots, s_N]$ making an acceptable tradeoff among throughput, phase offset and fairness. In Section 3.4.2, allocating channels proportionally to the popularities (NVoD T-PROP) is shown to be a good candidate for very impatient customers. For moderately to very patient customers, allocation in proportion to the square root of the popularities (NVoD T-SQRT) is preferable. Thus, we consider the *NVoD T-PROP* partition for $\beta = 0.01$ and *NVoD T-SQRT* for $\beta = 0.1$.

4. A QVoD server with the same partition $[s_1, \cdots, s_N]$ determined by NVoD T-PROP for $\beta = 0.01$ and NVoD T-SQRT for $\beta = 0.1$, used in conjunction with $\overline{K}^{opt}$. These configurations are called *QVoD-enhanced NVoD T-PROP* and *QVoD-enhanced NVoD T-SQRT*.

5. An NVoD server with $[s_1, \cdots, s_N]$ maximizing the NVoD throughput; this configuration is called *NVoD T-OPT*.

Figures 3.11, 3.12 and 3.13 show the simulation results for 100 channels partitioned among 10 movie titles of 100 minutes each, and accessed by very impatient customers ($\beta = 0.01$). We measured throughput, APO and dispersion. The improvement in throughput by using a QVoD server (configurations QVoD T-PROP and QVoD EW-OPT) is dramatic, with a relatively minor effect on the APO and the corresponding dispersion. This is particularly noticeable for large traffic intensities. For very low traffic intensities ($\rho < 1.0$),

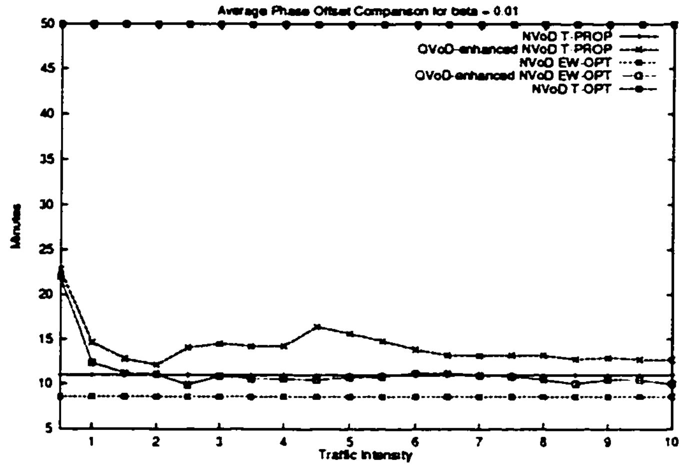


Figure 3.12: Comparison of NVoD and QVoD-enhanced NVoD: APO for $\beta = 0.01$.

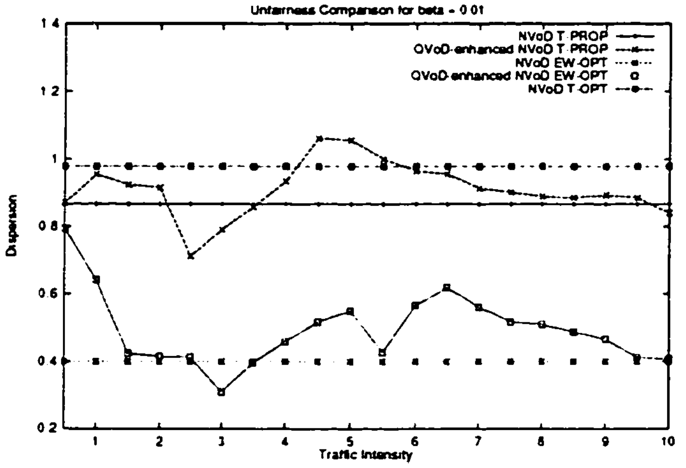


Figure 3.13: Comparison of NVoD and QVoD-enhanced NVoD: dispersion for $\beta = 0.01$.

QVoD EW-OPT is preferable to QVoD T-PROP, since it achieves a comparable throughput for lower phase offsets and dispersion. In summary, for very impatient customers, the configuration QVoD EW-OPT is the best choice, as it improves upon the maximum throughput achievable with NVoD (configuration NVoD T-OPT), for an APO comparable to that of the NVoD configuration NVoD T-PROP, and the corresponding dispersion comparable to that of NVoD EW-OPT. For more patient customers ($\beta = 0.1$) though, we found in separate simulations that the marginal improvement in throughput by replacing NVoD by QVoD is not worth the significant increase in the APO and dispersion. Finally, similar conclusions were made in the case of nonstationary arrival rates, for which QVoD can be used for very impatient customers.

3.7 Conclusions

In this chapter we first improved upon existing work on NVoD systems in several ways. First of all, we presented an analytical approach to program scheduling in NVoD systems, as opposed to commonly-used simulation such as in [12], and derived closed-form for various key performance variables such as the NVoD server throughput. By using a similar patience model, we extended the seminal work of [44], in which it is assumed for simplification that customers' requests are granted only if they agree to wait for exactly one phase offset. Although our analysis can be seen as an alternative to that presented in [31], we provided more insight into the resource allocation problem in NVoD by integrating both customers' and service provider's points of views in a mathematical framework. More particularly, we were able to determine, in linear time, the optimal schedule of movies of different popularities for maximum throughput and the lowest APO. These results were also generalized to the case of nonstationary request arrival rates. Further, we showed that in practice, the choice of a scheduling algorithm will only depend on the number of channels available, customers' patience, and on the performance variable which is most valued by the NVoD service provider, i.e., throughput, APO, fairness, or a tradeoff among all of these. In the latter case, simple heuristics such as the allocation of channels in proportion to movie popularities are found to yield good results. These heuristics may be used as an alternative to optimal schedules which have been found non-schedulable. In summary, we were thus able to (1) bound the system performance with optimal partitions, and (2) provide ad-hoc guidelines on resource allocation.

Next, we presented variations on the basic NVoD scheme. In QVoD, channels are scheduled based on a threshold of requests. The throughput was found to be usually greater in QVoD than in NVoD, except for the extreme case of nonstationary request arrival rates. This last observation was used to improve throughput without compromising customers' QoS, by using QVoD in conjunction with NVoD. This result is important since as outlined in Section 3.2, cost-effective NVoD scheduling requires time-consuming heuristics, and there is still no theoretical bound on the worst-case performance of NVoD retrieval scheduling heuristics in HCGS. A VoD server can delay channels until an available slot opens up, or source channels quasi-regularly as in QVoD-enhanced NVoD and yet provide a higher throughput than in NVoD, as long as each movie title is allocated the pre-determined number of channels. This fortuitous result may also be particularly useful when the popularity profile changes throughout the day, as it allows for more freedom in program scheduling, which can modify the movie retrieval periods within prespecified bounds.

Throughout this chapter, we have looked at particular combinations of batching policies and partitioning heuristics that minimize the loss of customers' QoS incurred by workload variations. These different approaches to scalable batching differ in complexity and schedulability. In the next chapter, we further investigate these issues by presenting a methodology to identify and compare scalable batching schemes in disk-array-based VoD servers. and applying this methodology to more general batching policies and storage organizations.

CHAPTER 4

SCALABLE BATCHING POLICIES
IN DETERMINISTIC VIDEO-ON-DEMAND SERVERS

4.1 Introduction

Chapter 2 dealt with the problem of specifying a low-cost storage organization for maximum customers' QoS while satisfying arbitrary constraints on channel capacity and movie selection. In Chapter 3, we then focused on monolithic storage organizations, and analyzed the commercially-available NVoD and variations on the basic scheme for increased throughput and scalability, and enhancement of basic D-VoD service to limited VCR functionality. In this chapter, we further investigate the general issue of achieving scalability in disk-array-based D-VoD servers by batching customers' requests.

As we have seen in Chapter 3, batching requests during the scheduling phase increases throughput and may decrease customers' average latency since requests for the same movie that arrived within the same batching period do not have to compete for resources. The choice of a batching policy thus plays an important role in determining both customers' latency and defection rate. The simplest and most popular batching policy, on-demand batching, batches outstanding requests *whenever a channel becomes available*. However, as we shall see, lack of regulation in the channel allocation process causes a considerable scalability problem during extended periods of demand surge, thus leading to *congestion cycles*. The main intent of this chapter is to provide a methodology to (1) identify non-scalable batching policies, and conversely, (2) evaluate and compare scalable batching heuristics that a D-VoD server, consisting of one or several disk arrays, can use in order to grant customers' requests with maximum QoS while achieving a high throughput.

4.1.1 Approach

A batching policy is scalable if it can accommodate possibly significant workload fluctuations and overloading situations without degrading customers' admission QoS, expressed in terms of the admission latency, defection rate due to long waits, and availability during reconfigurations. In addition to time variations in the request arrival rate, two factors have to be considered for their possible impact on scalability: (1) customers' willingness to wait (thus delaying their requests for batching), that is, *customers' patience behavior*, and (2) the storage organization of the D-VoD server. First, customers are only partially patient, and may thus renege service if they wait too long. As illustrated in Chapter 3, this reneging behavior imposes constraints on batching, whose key principle is to delay customers' service as long as possible for the best scalability. Second, in a clustered configuration, each cluster serves as a repository for a subset of all movie titles, for which it will provide a fraction of the total channel capacity. Clearly, the underlying storage organization will determine the maximum number of channels that can be allocated to a particular movie title and how the concurrent channel capacity is shared among various movie titles. It will therefore dictate the batching performance and the feasibility of a batching schedule. Also, when resources are (partially) shared, an admission scheduling policy may have to be used in conjunction with the batching policy in order to decide which movie title should be given priority. It is therefore important to evaluate the effectiveness of batching policies within representative storage organizations, or alternatively, the impact of a storage organization on the scalability of a batching policy.

In summary, we extend Chapter 3 in several ways. By analyzing a non-scalable batching policy, we introduce a scalability metric that will allow us to comparatively evaluate various scalable batching policies, including NVoD and related schemes. In parallel with this study, we propose scalable ad-hoc alternatives which are applicable in a wide range of clustered storage organizations, and offer an interesting tradeoff between customers' QoS and server's throughput.

The rest of this chapter is organized as follows. Section 4.2 gives background on several alternatives to batching that have been proposed in the literature for increasing the number of concurrent channels beyond the capacity limitations of available resources. We identify the scalability shortcomings of these resource sharing techniques. Next, we analyze the congestion problems associated with on-demand batching. In particular, we introduce a new metric called the *burst absorption capability* (BAC). We use this metric to comparatively review several well-known and ad-hoc scalable batching policies assuming infinitely- and partially-patient customers. We examine next the loss of customers' QoS that enables disk arrays to be clustered to provide customers with higher service availability. We will actually show that even the simplest scalable batching policy vastly improves on-demand channel allocation, while being easily implementable in both monolithic and clustered disk arrays. The chapter concludes with some remarks in Section 4.7.

4.2 Resource Sharing in VoD Systems

This section briefly reviews several techniques which have been proposed to achieve a high throughput and limited scalability in a VoD server. They attempt to share the resources that need to be reserved for real-time retrieval and delivery of video data from the storage repository to viewers. At the local VoD server level, these resources usually comprise (but are not restricted to) buffer space, storage bandwidth, and network bandwidth.

4.2.1 Caching

The *caching* (also called *bridging* or *buffering*) approach [30, 33] saves storage bandwidth by using buffer space in the VoD server memory to retain moving portions of certain videos as they play. This way, the data fetched by a client can be reused by others closely following the client, if memory space is available to retain the blocks. In general, depending on the buffer location in the video delivery path, caching trades extra memory for reduced bandwidth demand. If a static memory block allocation is used, the effectiveness of caching will depend on the intervals between two successive requests for the same video object. For low request arrival rates, a considerable amount of buffer space may be required in order to achieve any significant saving in server capacity. Thus, static caching is cost-effective in the restricted case of high request rates, and for frequently-requested videos. In this case, however, a significant amount of RAM may be needed unless the request rates are so (unrealistically) high that the data prefetched during retrieval in disk arrays (e.g., 2 seconds' worth of video in a 1,000-channel CGS disk array, and 24 seconds in a FGS disk array) may be immediately reused.

Additional optimization schemes such as *pinned demand paging* [78] and *interval caching* [33] attempt to exploit the sequential access to video data through dynamic caching of video data according to changes in the movie access frequencies. Nevertheless, as documented in [30], adaptive techniques are not scalable since even though a higher system capacity can be provided, customers' blocking probability increases linearly with the request rate. In the extreme case of saturation, all video segments have to reside in RAM, which, as noted in Section 2.1 of Chapter 2, is not cost-effective due to the size of video objects (e.g., storing a 100-minute movie requires 1.125 GB if MPEG-1 is used, and 5 GB with MPEG-2).

4.2.2 Adaptive Piggybacking

Unlike caching, using *multicast* communication to serve several customers with the same channel saves storage bandwidth, buffer space, and network bandwidth altogether, and hence lowers per-user costs. An alternative technique to batching, called *adaptive piggybacking*, was studied in [9, 58] to achieve multicasting "on-the-fly," through alteration of the display rates while the requests are in progress. This technique differs from batching, in which customers' requests are grouped at the beginning of a movie. The basic idea behind adaptive piggybacking is that if two streams for a common video are a small enough interval apart, the first stream can be played at a slower rate than the second; ultimately, the second stream catches up and can be piggybacked on the first. This is based on the fact that small differences in the display rates are not perceived by the viewer. In addition to difficult implementation issues when MPEG-like compression is used (because of inter-frame dependencies), one major problem with piggybacking is that the size of the catch-up window is severely constrained by the maximum acceleration rate permissible by QoS considerations. Thus, no resources are freed during the period of the catch-up window, which can be significantly long.

To alleviate this problem, [58] advocates content insertion (e.g., advertisements, previews) based on intermediate caching of alternate programs. Used in synergy with rate adaptive stream merging, this technique allows resources to be freed temporarily and used to decrease the catch-up window, and consequently, to accommodate minor overload situations. Short-term failures can also be dealt with easily if customers can be switched to alternate programs until resources become available. In spite of all these features, the authors of [58] note that content insertion may cause "stream-thrashing," in which service scalability is degraded by an increased amount of inserted content. Also, even though content insertion may subsidize the cost of entertainment provided to the consumer, it is not clear whether such a practice will not drive customers away from digital video services.

It should be noted that the schemes presented in this section are not mutually exclusive. For instance, adaptive piggybacking can be used in conjunction with batching to recover resources during unexpected short-term transients. As for interval caching, schemes presented in [79, 80], when used in conjunction with batching, can provide viewers with limited

VCR capability if large buffer costs can be afforded. These two approaches are therefore orthogonal to batching, and can be incorporated as additional optimizations. In the case of caching, however, it has been mentioned in [41] that with current hardware pricing and stream parameters, trading memory for disk bandwidth is often a losing proposition. Nevertheless, we will show in the next chapter that decentralized caching of portions of videos in CPE buffers is an interesting alternative to distribute the cost of providing *unrestricted* VCR functions without compromising the VoD service scalability achieved through batching. Lastly, one can observe that both batching and adaptive piggybacking with content insertion rely on assumptions made on viewers' tolerance. However, whereas the latter approach assumes tolerance to alteration in both rate and content of video programs, the former is based on the sole assumption that customers are ready to wait for a more scalable. unaltered service. As we shall see in the remainder of this chapter. batching requests for the same title allows a D-VoD server to facilitate heavy request loads. which would otherwise cause severe degradation in throughput and customers' waiting times.

4.3 On-Demand Batching in Disk-Array-Based D-VoD Servers

4.3.1 Scheduling

A straightforward approach to batching, known as "on-demand" (OD) batching, is to service outstanding requests together whenever channels become available. without any control in the channel allocation process. In order to be applicable to disk-array-based D-VoD servers, a *scheduling policy* has to be chosen to decide which movie title should be allocated to the first available multicast channel. Once a movie title has been chosen. batching is done by transferring corresponding newly-arrived requests from a request list to one or several service lists and by reserving the resources needed to start sourcing the program. Thus, customers' waiting time is affected by the scheduling policy used by the D-VoD server to transfer a request from the request list to the service list.whenever a channel becomes available. Several on-demand policies have been proposed and evaluated using simulation models [10]. These policies differ in the choice of the scheduling policy used:

FCFS (first-come first-served) batches all requests for the movie title requested first in the request list.

MQL (maximum queue length) batches all requests for the movie title with the largest number of requests in the request list.

MFQL (maximum factored queue length) [10] schedules the video with the largest factored queue length, i.e., the queue length for a video divided by the square root of its popularity. It was proved in [10] that MFQL ensures minimum waiting time for infinitely (in some cases, partially) patient customers. MFQL is also a fairer policy, when compared to MQL in particular, in that it reduces the dispersion of latencies from a movie title's point of view. Hence, movie titles are less likely to suffer from starvation caused by unfair scheduling policies.

LCFS (last-come first-served) batches all requests for the movie title requested last in the request. LCFS is known to be better suited to *very* impatient customers [100].

As noticed in [41], one of the problems with on-demand batching is that unlike NVoD, it is unpredictable, in the sense that it cannot offer a guaranteed upper bound on how long a client request must wait in the admission queue. Also, results on the behavior of such policies (either analytical or by simulation) are typically based on specific probabilistic models of customers' reneging behavior, whose accuracy is often questionable in practice. More importantly, most movies are known to have playback lengths of about 90 – 100 minutes, usually well-approximated by uniform or Gaussian distributions. Due to lack of variability, a large number of channels may be released in a short time, and then immediately reallocated. In case of sustained high request loads, such phenomenon, referred to as *channel clumping*, causes the D-VoD server to remain unavailable for most of the time by "memory effect", and both customers' QoS and server throughput will degrade. This phenomenon is exacerbated by (1) the heavy-tailed [29] Zipf-like localized access to these movies, and (2) the disparity discussed in Section 2.3 of Chapter 2) between the number of individual movie titles usually held in most storage organizations and the much higher concurrent channel capacity allocated to these movies. This sensitivity of customers' QoS to channel clumping may also be found in more general "greedy policies" which allocate channels on an on-demand basis. An example of such policies is the *forced wait* policy in [11, 87], in which the request at the head of the request queue is required to wait some pre-determined minimum amount of time before being serviced.

4.3.2 Congestion Cycles

Channel clumping occurs at unpredictable times, and then disrupts customers' QoS for extended periods of time. To illustrate this point, we considered a CGS disk-array that provides access to N different movie titles whose length is either constant, or uniformly distributed between 80 and 120 minutes, with an average length of $L = 100$ minutes in both cases. We assume Poisson request arrivals with parameter λ, a channel capacity of s channels, and Zipf-like movie popularities $\bar{p} = [p_1, p_2, \cdots, p_N]$. The traffic intensity is given by $\rho = \frac{\lambda}{s\mu}$. Note that from a customer point of view, the average movie length can be defined as $L_c = \frac{\sum_{i=1}^{N} L_i}{N}$, whereas from the D-VoD server point of view, the average duration of a channel is not known a priori (it is $L_s = \sum_{i=1}^{N} p_i L_i$ if channels are fairly shared among titles).

General arbitrary models of time-varied request arrival patterns have been proposed in [11]. These models comprise successive intervals with approximately constant request arrival rates over an extended period of time (e.g., 1 hour). As mentioned in Chapters 2 and 3, considering the lack of published realistic models of customers generating non-static requests in a real VoD system, workload characterization is necessarily controversial. Nevertheless, we chose here a model which would, ideally, reproduce a realistic *dynamic* demand on the D-VoD server while displaying the shortcomings of on-demand batching. We thus considered an arrival model, denoted by $(\bar{\rho} = [\rho_h, \rho_m, \rho_l], RP, T)$, of period T, alternating between plateaus of high and medium traffic intensity (ρ_h and ρ_m), interleaved with plateaus of traffic intensity randomly distributed between 0 and ρ_l. Variations between consecutive plateaus are linear, and the plateau width is randomly distributed between $RP \cdot \frac{T}{4}$ and $\frac{T}{4}$. With very few parameter adjustments, a wide array of "worst-case" nonstationary demand patterns can be generated as successions of peak periods with uneven durations and amplitudes and continuous variations in between. This model also captures *continuous* customers' cyclic behavior. For instance, a plateau width greater than 5 hours and a 24-hour period can represent viewers' demand peaking from 7 p.m. to midnight, and with moderate load from 7 a.m. to noon. Also, plateaus of low traffic intensity, during which few channels are requested, make the system prone to channel clumping and congestion cycles.

Figures 4.1 and 4.2 represent, over a one-week period, generated $([20.0, 10.0, 1.0], 0.8, 24)$ and $([10.0, 5.0, 1.0], 0.8, 24)$ workloads with a uniform plateau distribution. For readability, the averaging period was set to 1 hour in Figure 4.1, and 10 minutes in Figure 4.2. We also assumed that at $t = 0$, all channels are free, and only allocated one after another at regularly spaced intervals of duration $\frac{L_c}{s}$ over the first period of length L_c. Clearly, this initial allocation pattern is not enough to prevent congestion cycles from building up after extended period of pseudo-stability (e.g., from hour 40 to 100 in Figure 4.1). These cycles span over several periods, then fade away before starting again. As documented in [11] on the time scale of a burst (e.g., 1 hour), channel clumping during high loads also causes

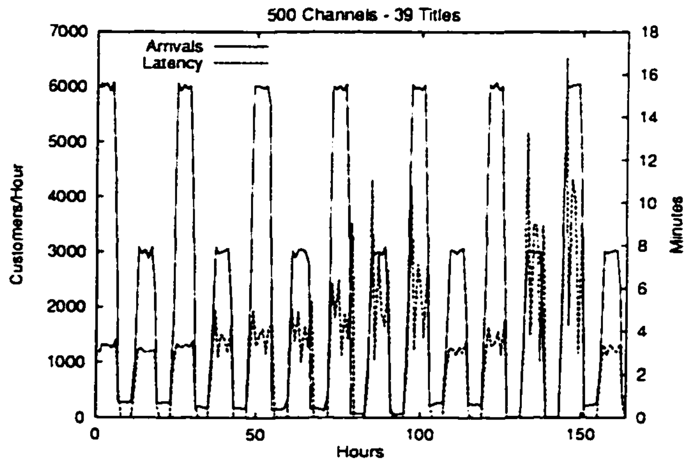


Figure 4.1: Congestion cycles in on-demand batching for ([20.0, 10.0, 1.0], 0.8, 24).

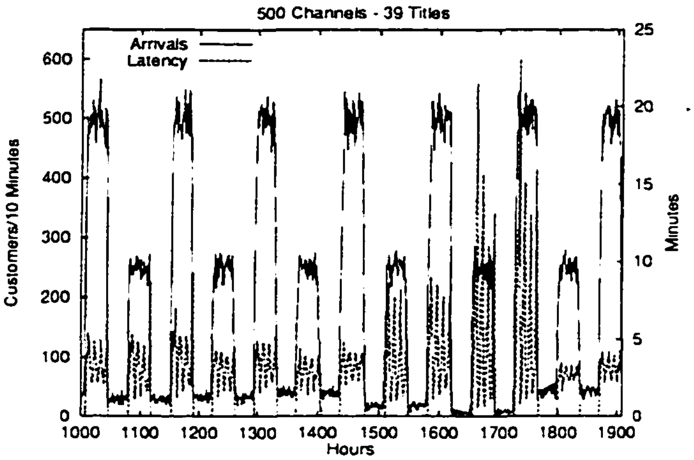


Figure 4.2: Congestion cycles in on-demand batching for ([10.0, 5.0, 1.0], 0.8, 24).

wide oscillations in latency, which occur with the same intensity during both peak and light load periods, hence irrespectively of the variations in traffic intensity. More importantly, however, the *cyclic long-range* behavior on the time scale of a day is also independent of the traffic intensity during plateaus. Lastly, although we have identified this congestion behavior with a specific ideal "worst-case" workload, in the general case, channel clumping may happen more or less frequently and then cause unpredictable congestion.

4.3.3 Burst Absorption Capability Analysis

Ideally, a scalable batching policy should always prevent the degradation of customers' QoS and server throughput. Performance should thus be as *self-similar* as possible, that

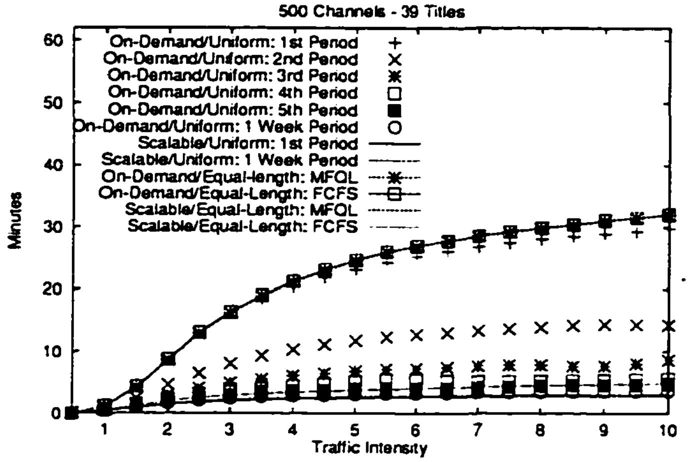


Figure 4.3: Burst absorption capability of on-demand batching.

is, remain approximately constant and acceptable on various time scales, regardless of the duration and amplitude of high load periods. This property guarantees a congestion-free batching policy, devoid of short- and long-term oscillation cycles. Our methodology to evaluate scalability consists of studying the BAC, which is defined as the time it takes for an initially-empty D-VoD server to recover from a sudden burst of requests at a given rate ρ, and of infinite duration. In practice, the BAC is measured by comparing customers' QoS averaged over successive periods of time.

As an example, we considered in Figure 4.3 the simulation of a D-VoD server with 39 videos of equal or uniformly-distributed durations, in a CGS disk array of channel capacity $s = 500$. We evaluated the BAC by measuring the admission latency averaged over successive periods of duration L_c each. In the case of on-demand batching with equal-length movies, customers' latency is always unacceptably high, regardless of the averaging period, and increases with ρ during the burst. (It is also interesting to note that although MFQL is superior in theory, this scheduling policy only yields a marginal improvement as compared with FCFS in the case of equal-length movies.) This result is not surprising since when at some time t, a large number of channels are available and suddenly requested at peak rate, the concurrent channel capacity of the D-VoD server will be saturated at once, for a period of time corresponding to the length of a movie. Once a large fraction of the channel capacity is released, the system is ready for another *congestion cycle*.

With uniformly-distributed movie lengths on the other hand, the variability in movie lengths allows the system to recover from such "self-entertained" bursts in five periods or approximately 8 hours. When channels are sourced at fixed-length intervals of duration

$\frac{L_c}{s}$ each, the latency remains low, regardless of the averaging period and ρ. Clearly, the BAC study showed the deficiencies of on-demand batching and identified a scalable batching policy, which will be elaborated on in the next section.

4.4 Scalable Batching in Disk Arrays

The focus of this section is on comparing the performance of various approaches to scalable batching in D-VoD systems, specifically chosen for their different implementation complexity. These policies achieve service scalability through *admission regulation*, by exploiting customers' tolerance to moderate waiting times within the batching period. Our approach is to first use the BAC analysis to compare the various policies. In parallel with our comparative study, we wish to show through appropriate "stress tests" that even the simplest scalable batching policy may vastly improve on-demand channel allocation.

4.4.1 Near Video-on-Demand

As presented in Chapter 3, batching in NVoD is to determine a program *schedule* according to which the same material is sourced at almost equally-spaced intervals, called *phase offsets*, if requests were placed during the last $\frac{L}{s_m}$ time units[1]. NVoD customers therefore experience a predictable latency which is independent of the traffic intensity. A variation of NVoD consists in completely sharing the channel capacity s within a same disk array among all movie titles and sourcing scheduled video programs every $\frac{b=\bar{L}}{s}$ units of time, where $\bar{L}$ is the average length of a program. Movie titles are selected according to one of the scheduling policies introduced in Section 4.3.1. Such a system is said to be *heterogeneous* NVoD (NVoD-h). In the case of equal-length movies, $\bar{L} = L$. In the case of uniformly-distributed movie lengths, $\bar{L}$ is calculated such that b corresponds to the average length of a busy period, that is, the average time interval between the availability of two consecutive free channels, when the VoD server is congested.

4.4.2 Rate Control

Rate control (RC) [11] is an ad-hoc technique to eliminate congestion by reducing variability in busy periods, thus always providing customers with the same channel availability. RC was shown to perform better than more general "greedy" (e.g., on-demand) batching

[1] We shall henceforth consider *non-work-conserving* NVoD systems, which provide a more efficient bandwidth utilization by not restarting service every $\frac{L}{s_m}$ units of time if no new service requests were placed in the previous $\frac{L}{s_m}$ time units.

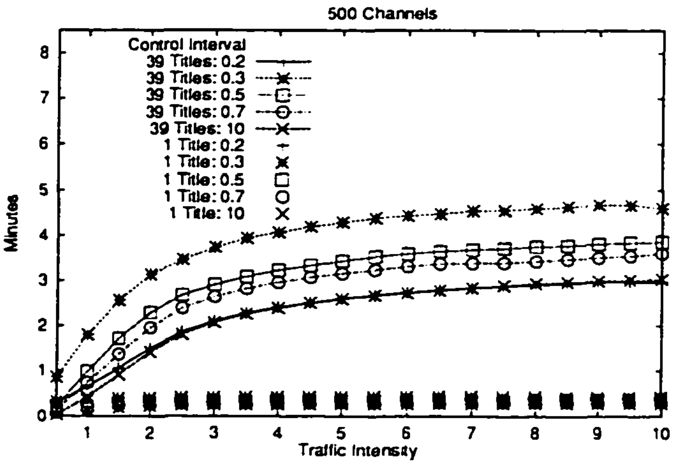


Figure 4.4: Performance of RC for various values of T_{int}.

policies. The basic idea behind RC is to allocate channels as uniformly as possible by specifying a control interval of $T_{int} \leq \bar{L}$ minutes during which the total number of channels allocated is upper-bounded by $s_{max} = \lfloor \frac{T_{int} \cdot s}{\bar{L}} \rfloor$. Then, channel allocation is controlled by enforcing a minimum interval Δ from one channel allocation to the next, which is computed at run-time as $\Delta = \frac{t_{left}}{s_{max} - s_{allocated}}$, where t_{left} is the time left until the next control interval, and $s_{allocated}$ the number of channels already allocated. Ideally, because s_{max} is an integer, T_{int} should be chosen in multiples of $\frac{\bar{L}}{s}$ for optimal results. In a D-VoD system with equal-length movies, if T_{int} is not chosen in multiples of $\frac{\bar{L}}{s}$, channels are allocated at a faster rate, thus resulting in moderate congestion and higher latency. This is illustrated in Figure 4.4, which shows the BAC graph for RC in a 500-channel system with a movie selection of 1 and 39 titles.

$T_{int} = \frac{\bar{L}}{s}$ corresponds to NVoD-h. One potential advantage of RC over NVoD-h is that if T_{int} corresponds to several busy periods, channels can be allocated on-demand, thus without any control, during very small bursts in a non-saturated system. As a result, customers are not forced to wait unnecessarily. In practice, the choice of an optimum control interval depends on the expected level of variations in the request arrival pattern. This feature is, however, of limited use when peak periods of unpredictable duration and high intensity happen randomly, since in order to provide a short response time before saturation, T_{int} should actually be chosen as small as possible. For instance, in the particular case of workloads presented in Section 4.3.2, T_{int} should be set to a few estimated busy periods depending on the accuracy of $\bar{L}$. This is illustrated in Figure 4.5 for a ([10.0, 5.0, 1.0], 0.8, 10) workload and uniformly-distributed movie lengths: peak periods may happen right before

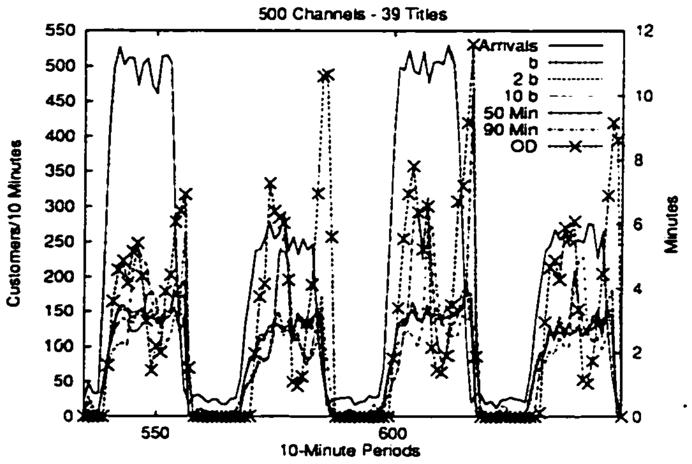


Figure 4.5: Choice of a control interval.

the end of an interval of duration T_{int}, during which very few channels have been allocated prior to the sudden burst of requests; as a result, RC degenerates to on-demand batching when T_{int} exceeds a certain limit. On the other hand, $T_{int} = n\frac{L_c}{s}, n = 2, 10$ provides results similar to NVoD-h ($n = 1$).

Similar results were obtained with the *deviated rate control*, which is a variation on the basic RC scheme (also called *pure* rate control) proposed in [11] more adapted to partially-patient customers generating sudden request surges in very short time periods (e.g., a few minutes). This is not surprising since deviated control was primarily introduced to prevent reneging during very short-term load surges (e.g., on the time scale of a few minutes), which are not present in our workload models. Although more work is needed to evaluate deviated rate control in more general nonstationary workloads, the general results displayed in Figure 4.5 should remain unchanged.

4.4.3 Quasi Video-on-Demand

QVoD systems, introduced in Chapter 3, regulate channel allocation according to the underlying traffic intensity, based on a threshold on the number of pending requests. As we have seen, NVoD can be upgraded to a QVoD system called *QVoD-enhanced NVoD* (or *QNVoD* in the figures presented in this chapter), by (1) partitioning the channel capacity s into $[s_1, \cdots, s_N]$ as in NVoD; and (2) defining a vector of optimal thresholds $\overline{K}^{opt} = [K_1^{opt}, \cdots, K_N^{opt}]$. QVoD-enhanced NVoD may yield a dramatic improvement over NVoD in terms of customers' latency and throughput. Similarly to the distinction between NVoD and NVoD-h, if the channel capacity s is completely shared by all titles, channels may

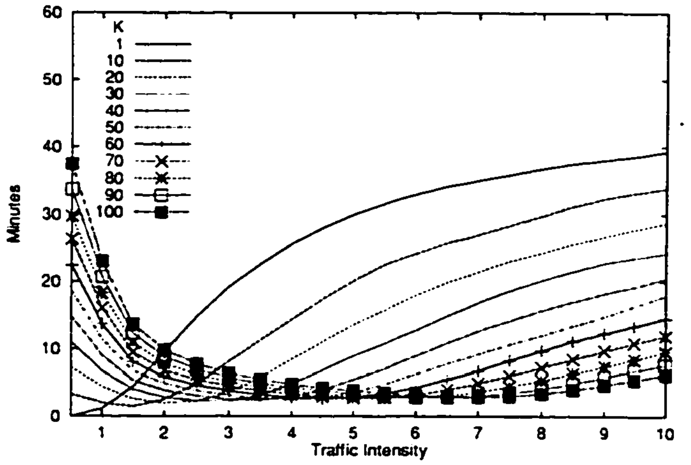


Figure 4.6: Average latency of threshold-based MFQL.

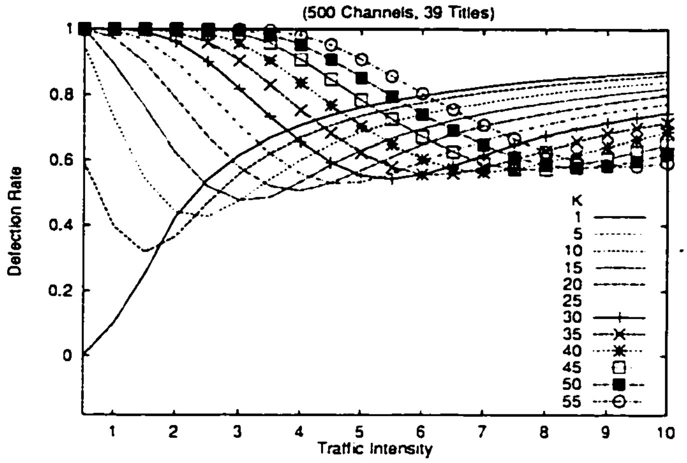


Figure 4.7: Defection rate for $\beta = 10$.

source scheduled movies only as soon as a certain threshold on the number of pending requests has been reached, and regardless of the titles requested. Such a system is said to be *heterogeneous* QVoD (QVoD-h). Note that on-demand batching is a special case of QVoD-h with $K = 1$.

Determining the best thresholds is usually intractable as they vary with the system operating point (e.g., traffic intensity, customers' patience). This can be seen in Figure 4.6, which represents the average latency experienced by infinitely-patient customers, and in Figure 4.7, which shows the defection rate of partially-patient customers. Nevertheless, when customers are infinitely patient, the first graph of Figure 4.8 illustrates for pure Zipf and uniform popularity distributions that K^{opt} is approximated well by the linear form $\lceil c_{\bar{p}} N\rho \rceil$, where $c_{\bar{p}}$ is a coefficient of variation which only depends on popularities. When

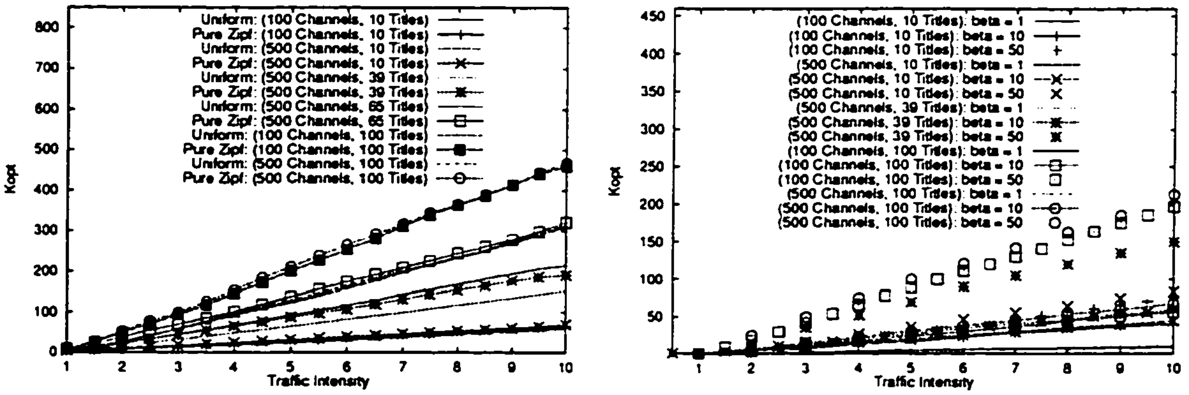


Figure 4.8: Variations of K^{opt} with infinitely- and partially-patient customers.

customers are partially patient, the choice of an optimal threshold depends on whether customers' latency or defection rate is more valued by the D-VoD server. As seen in Chapter 3, there is an optimal value K^{opt} that minimizes customers' defection rate. This optimal value does not necessarily minimize customers' latency, which is calculated based on requests that actually get serviced. In general, however, servicing a large number of customers is more important than providing a low-latency service at the cost of a high defection rate. It is therefore reasonable to choose a threshold that minimizes the defection rate. As can be seen in the second graph of Figure 4.8, K^{opt} also varies linearly with the traffic intensity for the customer's patience model introduced in Section 2.6.2 of Chapter 3. In practice, we can plot K^{opt} in graphs similar to Figure 4.8 and dynamically adjust K to K_{opt} corresponding to the instantaneous arrival rate of requests. If the instantaneous arrival rate is not known, a statistical change detection algorithm may be used on-line to detect intervals with stationary request arrival rates and adjust the threshold accordingly. An example of such techniques is given in [50].

In summary, the various scalable batching policies can be classified in terms of complexity, from QVoD-enhanced NVoD and QVoD-h which require significant tuning, to NVoD-h which requires no tuning at all. To a lesser extent, NVoD also requires very little adjustment considering that the various partitioning heuristics presented in Chapter 3 run in linear time.

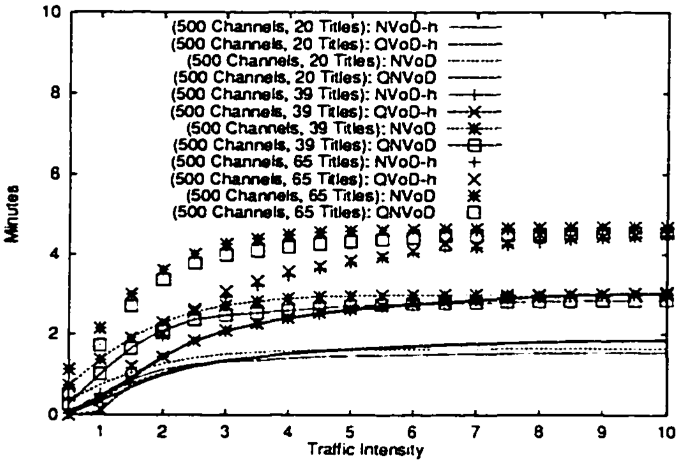


Figure 4.9: BAC of scalable batching policies for 500 channels.

4.5 Performance Comparison

4.5.1 Infinitely-Patient Customers

We first investigate the BAC in Figure 4.9 for a movie selection of 20, 39, and 65 titles, and infinitely-patient customers. In NVoD and QVoD-enhanced NVoD, the partition $[s_1, \cdots, s_N]$ was chosen so the average phase offset in NVoD is minimized. As the movie selection increases, NVoD-h performs better than other policies. For small movie selections (less than 20 titles), QVoD-enhanced NVoD is optimal. For such low selections, however, it is clear that the average latency is so low that QVoD-enhanced NVoD only marginally improves upon other policies. Similar observations were made in other cases of channel capacity, ranging from 100 to 1,000. As illustrated in Figure 4.10 for a ([10.0, 5.0, 1.0], 0.8, 24) workload, NVoD-h with MFQL scheduling provides a remarkably simple way to achieve scalability and low, uniform latency for realistic storage parameters.

4.5.2 Partially-Patient Customers

Here, we consider partially-patient customers whose behavior was introduced in Section 3.3.1 of Chapter 3. Further, we define the *relative patience factor* as $\beta = \frac{\bar{\tau}_s}{L}$. This metric which indicates customers' average willingness to wait for service *relative to the server average busy period*, is representative of customers' patience relative to the VoD server channel capacity, since defections only occur during busy periods. In addition to NVoD-h, QVoD-h, and on-demand batching, we considered the following NVoD and QVoD-enhanced NVoD systems:

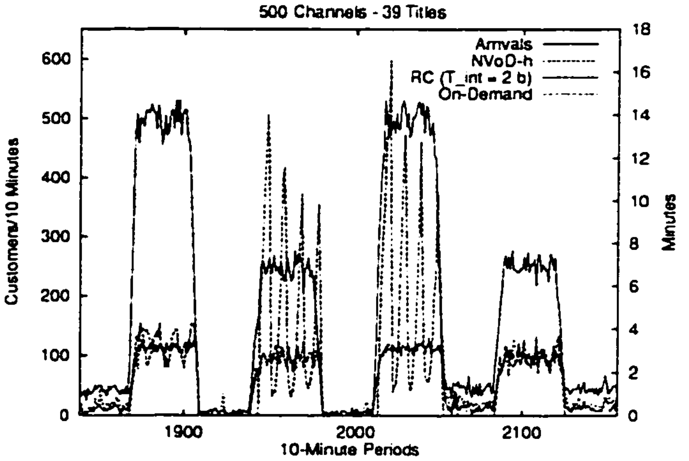


Figure 4.10: Congestion cycles in scalable batching policies for ([10.0, 5.0, 1.0], 0.8, 24).

1. An NVoD server with $[s_1, \cdots, s_N]$ minimizing the average NVoD phase offset; this configuration, called *NVoD EW-OPT*, also yields minimum latency in the case of infinite patience, by allocating channels more evenly among movie titles.

2. A QVoD-enhanced NVoD server with the same channel allocation vector $[s_1, \cdots, s_N]$ defined by NVoD EW-OPT, used in conjunction with $\overline{K}^{opt}$ which minimizes the defection rate for each movie title. This configuration is called *QNVoD EW-OPT*.

3. An NVoD server with the partition $[s_1, \cdots, s_N]$ minimizing the defection rate (or equivalently, maximizing the throughput). This configuration, called *NVoD T-OPT*, indicates the maximum throughput achievable by an NVoD server. NVoD T-OPT tends to allocate all channels to the most popular movies as the patience factor decreases.

4. A QVoD-enhanced NVoD server with the same partition $[s_1, \cdots, s_N]$ determined by NVoD T-OPT. This configuration is called *QNVoD T-OPT*.

These policies determine the range of defection rates and latencies achievable by NVoD and QVoD-enhanced NVoD systems with partially-patient customers. In practice, other partitioning heuristics such as allocating channels proportionally to the popularities, perform somewhere in between T-OPT and EW-OPT, thus achieving an acceptable tradeoff between defection rate and phase offset.

In Figure 4.11, we compare the BAC in the defection rate and average admission latency for equal-length movies. In our simulations, we considered $\beta = 10$ and 50, which, for $s = 500$ and $L = 100$ minute, correspond to customers who are willing to wait an average of

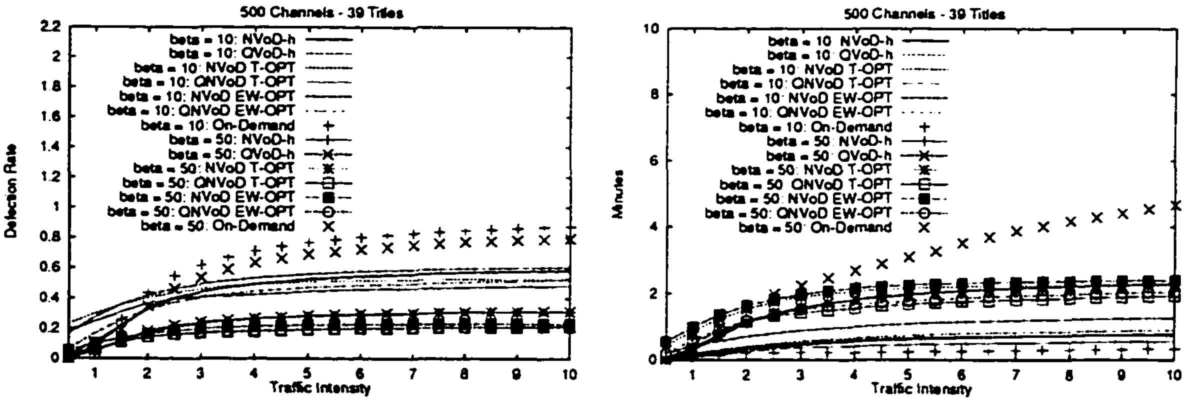


Figure 4.11: Defection rate and average latency with partially-patient customers for 500 channels.

2 minutes (thus relatively impatient) and 10 minutes (thus moderately patient). Note that these results are independent of the averaging period since all policies are scalable, with the exception of on-demand batching. Although all these batching policies seem to be quite similar to each other, we actually found that QVoD-enhanced NVoD policies increasingly outperform all other systems as channel capacity and movie selection decrease, but at the cost of extensive tuning. This result is illustrated in Chapter 5 for a 200-channel capacity.

In the case of impatient customers ($\beta = 10$), simulations indicate that scalable batching policies fail to prevent the high defection rate and wild oscillations observed in on-demand batching. In addition, NVoD and NVoD-h tend to force customers to wait for a duration which is likely to exceed their willingness to wait. For low traffic intensities, customers are thus forced to wait unnecessarily in a non-congested system, and on-demand batching may be preferable. In fact, preventive discarding of customers' requests *only during congestion periods* proved to reduce latency without noticeably affecting the overall defection rate. This case, however, is not representative since high defection rates should be avoided at all cost.

In summary, although it is not possible to detail here each particular case of channel capacity, movie selection, and customers' patience, all our simulations confirmed that NVoD-h with MFQL scheduling provides a remarkably simple way to achieve scalability, low latency and defection rate for realistic storage parameters. This is illustrated in Figure 4.12, which represent short-term defection rate and latency for a ([10.0, 5.0, 1.0], 0.8, 24) workload and $\beta = 50$.

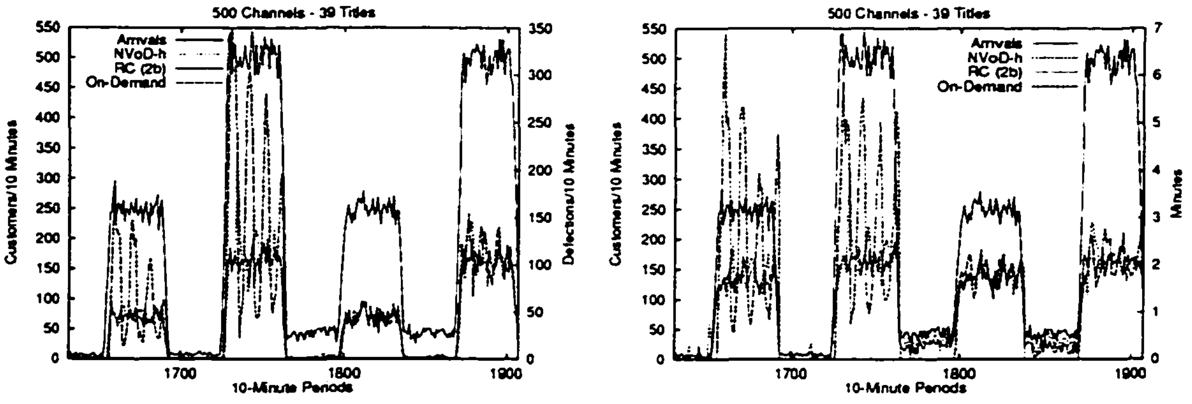


Figure 4.12: Defection rate and latency in scalable policies for ([10.0, 5.0, 1.0], 0.8, 24).

4.5.3 Discussion

A system with relatively impatient customers and a high defection rate is not economically viable. One should, therefore, use field studies to determine realistic values for $\bar{\tau}$, the parameter introduced in Chapter 3 to indicate customers' average willingness to wait. To illustrate the importance of such studies, we introduce here the *normalized effective cost* (NEC), which corresponds to the storage cost per channel and per served customer. The NEC can be calculated as $\frac{C_s}{s(1-D_r)}$, where C_s is the cost of the storage organization, and D_r the defection rate. For a given channel capacity, a lower NEC indicates a more efficient storage utilization, and therefore, a more profitable system.

In case of a high defection rate, a service provider has the choice between reducing the movie selection, increasing the server capacity, or doing both. The first approach is illustrated in Figure 4.13, in which the NEC is plotted in NVoD-h for 500- and 1,000-channel capacities as a function of the selectivity of videos offered to customers, parameter introduced in Section 2.4.2 of Chapter 2. (Selectivity 0 corresponds to a single movie title, whereas selectivity 1 corresponds to the maximum movie selection.) We considered five $\bar{\tau}$ values, ranging from 12 seconds to 20 minutes, and assumed a fixed arrival rate corresponding to $\rho = 10$ for each channel capacity. Clearly, reducing the number of movie titles has a minor effect on the NEC.

The second approach is illustrated in Figure 4.14, which plots the variations of the NEC as a function of the channel capacity in two cases of movie selection: (1) the maximum movie selection possibly held in storage, and (2) a fixed selection set to 39 titles. As can be seen, increasing the channel capacity is only advantageous for small capacities (e.g., from 500 to 1,000) when customers are very impatient ($\bar{\tau}$ = 12 seconds): in this case, the NEC decreases

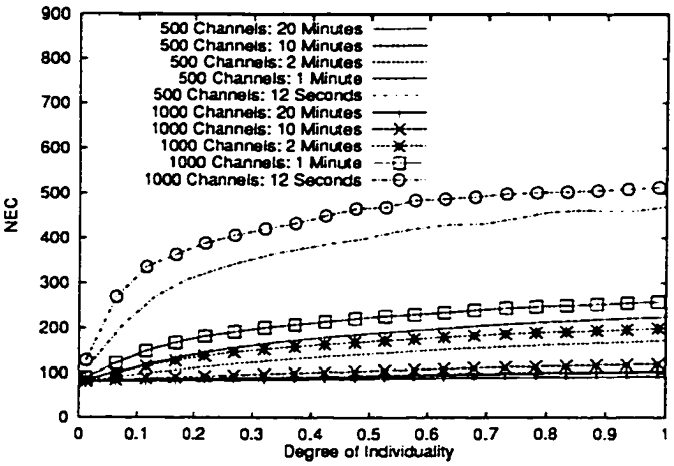


Figure 4.13: Normalized effective cost for various movie selections.

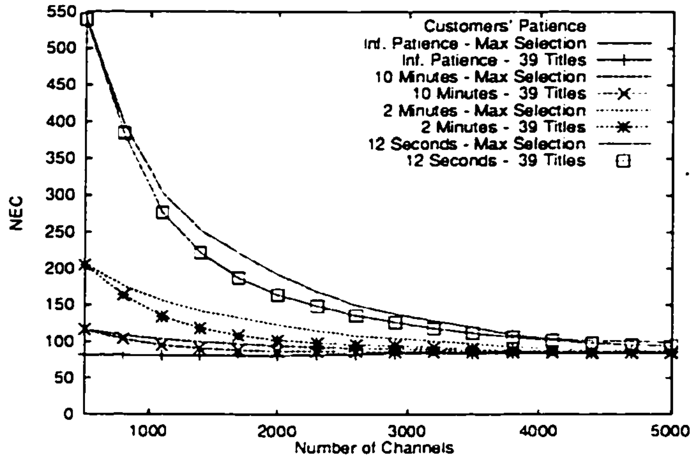


Figure 4.14: Normalized effective cost for various channel capacities.

at the same rate (from $550 to $275) as the channel capacity is increased. Consequently, if customers have been found very impatient, defections can be reduced in a profitable way. This approach is much less effective in all other cases of patience and channel capacity. In summary, our results emphasize that careful evaluation of customers' willingness to wait relative to the system capacity is of paramount importance when dimensioning disk-array-based VoD servers for economical viability.

4.6 Scalable Batching in Clustered Disk Arrays

4.6.1 Motivation of Ad-Hoc Batching

As presented in Section 2.4 of Chapter 2, video material renewal and content insertion of advertisements or movie previews require daily or weekly storage reconfigurations. During these reconfigurations, *narrow striping* across multiple independent *clusters* should be used to minimize service disruptions when changes occur. In clustered disk arrays, customers' QoS can be seen as a combination of (1) scalability, and (2) *service availability* during reconfigurations. The latter property is critical in a multicast VoD server, in which, typically, a large number of customers may be affected by service disruptions.

Once a partition $[s_1, \cdots, s_N]$ has been determined in a clustered NVoD or QVoD-enhanced NVoD system with C clusters, each movie title i requires $\lceil \frac{s_i}{B_c} \rceil$ clusters, where B_c is the bandwidth of one cluster in number of channels. Consequently, the same number of slots should be available for replication of this particular movie title. This constraint may considerably restrict the feasibility of optimal schedules in both NVoD and QVoD-enhanced NVoD systems. This is clearly illustrated by considering a 13-cluster 500-channel CGS disk array with 39 titles. Since there is no room for replication, the most popular movie title is limited to 39 channels, which is sub-optimal in both T-OPT and EW-OPT channel partitions when $\beta < 50$. In general, determining the feasibility of optimal schedules in clustered disk arrays and replicating videos accordingly are quite problems which depend on movie selection, channel capacity, number of clusters, and customers' patience. These issues are further complicated by the complexity of cost-effective NVoD scheduling in monolithic disk arrays (cf. Section 3.2 of Chapter 3), in terms of (1) determining NVoD-schedulability, and (2) laying out data based on the knowledge of phase offsets, which depend on — possibly time-dependent — popularity profile and customers' patience behavior.

For all these reasons, we focus on the ad-hoc NVoD-h in clustered disk arrays, for its implementational simplicity is a valuable asset to VoD service providers experimenting with feasible, scalable, yet reliable multicast VoD. Clearly, if an optimum NVoD or QVoD-enhanced NVoD schedule is feasible — as in some constraints located in area **C** — the performance of NVoD and QVoD-enhanced NVoD remains unaffected by storage partitioning. In NVoD-h however, channels are sourced at equally-spaced time intervals within a same cluster and regardless of the movie title. Consequently, increasing the number of clusters may directly affect the batching performance, since depending on the space available for replication, the maximum number of channels that may be potentially allocated to a

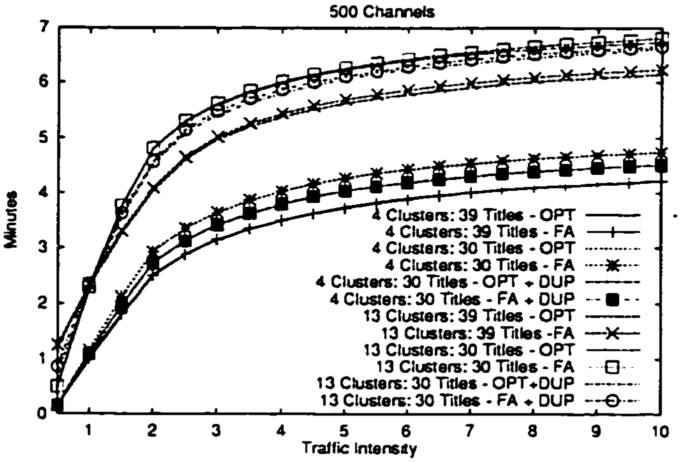


Figure 4.15: Average latency of NVoD-h for various assignments of videos to clusters.

particular movie title may decrease as well. In the rest of this section, we shall evaluate the effect of clustering in the most extreme case, that is, when no video replication across clusters is possible.

4.6.2 Video Allocation in Eligible Configurations

As a preliminary step in this section, we evaluate the algorithm for optimal video allocation presented in Section 2.4.2 of Chapter 2. More precisely, we wish tó determine whether optimal allocation is as critical in the case of NVoD-h as it is in a non-multicast D-VoD. In Figure 4.15, we plot for 4 and 13 clusters (1) the greedy algorithm (FA) vs. optimal video assignment in the case of 39 movie titles; (2) FA vs. optimal video assignment in the case of constraint A; and, (3) proportional replication vs. duplication (DUP) with respect to the admission latency in the case of constraint A. The figure shows that FA is sufficient to provide near-optimal allocation of videos to clusters, since, even in the extreme case of a 13-cluster configuration, once can only observe a marginal discrepancy with the optimal allocation. This observation contrasts with the non-multicast case, in which even a slight difference in the dispersion of access frequency across clusters between greedy and optimal assignments was shown to result in a noticeable discrepancy in the overall admission latency and sustainable rate. In addition, whenever storage space is available (as in constraint A), the choice of a replication policy only slightly affects the average latency as the number of clusters increases, and simple duplication is acceptable.

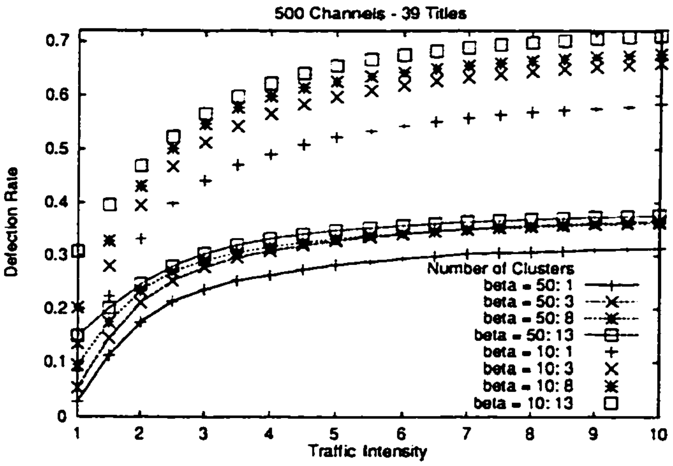


Figure 4.16: Defection rate for 500 channels and 39 titles.

4.6.3 Simulation Results

As we have seen in Chapter 2, a high availability can be achieved even with moderate partitioning. Since only non-replicated videos are not available during reconfigurations, a larger number of smaller clusters is attractive as fewer customers will be affected by reconfigurations. Smaller clusters may also be reconfigured faster. Excessive partitioning, however, tends to cause additional defections since customers become more impatient relatively to the average busy period in each cluster. This observation holds for both replicated and non-replicated videos, although it is more critical in the latter case because the maximum number of channels that can be allocated to a non-replicated movie is constrained by the cluster capacity. Fortunately, for moderately patient customers accessing a 500-channel CGS array storing 39 titles, Figures 4.16 and 4.17 indicate that as the number of clusters increases, the additional defection rate and latency in NVoD-h is insignificant in view of its high availability. This is not surprising since NVoD-h is relatively insensitive to variations in traffic intensity and channel capacity. In addition, scalability ensures that only those customers requesting non-replicated movies are affected. Note that in the case of $\beta \leq 10$, the defection rate ($> 50\%$ for $\rho > 4$) is high regardless of the number of clusters. As seen in Section 4.5.2 however, this extreme case is unlikely in a well-dimensioned VoD server.

4.7 Conclusions

Based on the results obtained in Chapters 2 and 3, the most original contribution presented in this chapter is a systematic and realistic comparison of various batching schemes

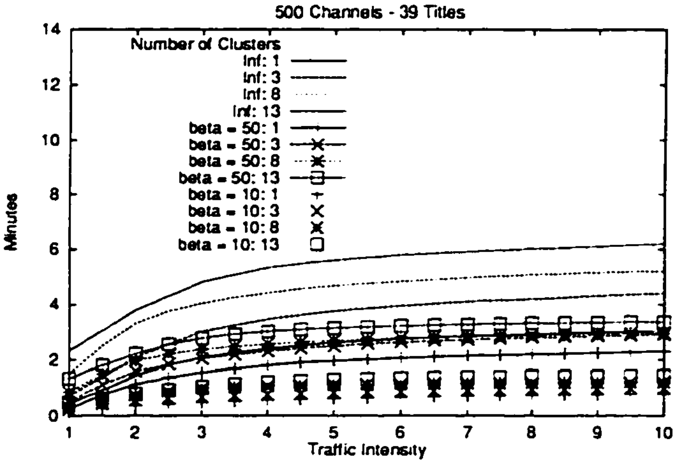


Figure 4.17: Average latency for 500 channels and 39 titles.

by considering factors such as customers' behavior, storage organization, and time variations in request arrival rates. We focused on disk-array-based D-VoD servers, whose movie selection and channel capacity are determined by cost and the striping scheme used. We presented on-demand batching policies and identified the lack of scalability, which causes short- and long-term degradation of customers' QoS and server throughput. Through our comparative study, we identified (1) the cause of congestion (channel clumping), (2) how to trigger congestion with specific workloads, and (3) how to measure congestion and scalability by considering the BAC. Based on these preliminary results, we then comparatively evaluated various scalable batching policies, showing that the simplest scalable batching policy, NVoD-h, may vastly improve on-demand channel allocation. We also identified the general limitations of batching when customers are impatient, that is, when their average willingness to wait is of the order of or lesser than the average busy period of the server. Finally, we extended the results to clustered disk arrays, and showed that high service availability and scalable batching can be achieved in a cost-effective way, even in worst-case situations where no movie is replicated.

Several simplifying assumptions made in this chapter regarding customers' behavior are purely speculative, and mostly justified for their intuitive realism. Our patience model can be further improved by considering customers with heterogeneous or nonstationary behavior, for instance, with several, possibly time-varying, patience factors. Also, the periodic workload models of Section 4.3.2 are somewhat arbitrary, and in reality, demands may not vary in a simple periodic fashion. Nonetheless, until field data confirms their pertinence, our assumptions and their proper are inevitable to (1) glean heuristics for efficient batching in D-VoD servers. and (2) understand the limitations of these heuristics when the (commonly-used) assumption of infinitely-patient customers generating stationary requests is relaxed.

We mentioned in Section 4.2 of this chapter and in Section 3.2 of Chapter 3 that partial caching of videos in the CPE buffer can be used to provide limited support for intermittent VCR actions in NVoD, and more generally, in multicast VoD systems. In the next chapter, we explore this issue further and present active CPE-buffer management and resource reclamation mechanisms for *unrestricted* VCR functions without compromising the VoD service scalability achieved through batching.

CHAPTER 5

PROVIDING UNRESTRICTED VCR FUNCTIONS IN MULTICAST VIDEO-ON-DEMAND SERVERS

5.1 Introduction

As we have seen in Chapter 3, some multicast VoD systems already provide customers with a framework for limited and scalable VCR capability. In NVoD and QVoD for instance, since video programs for the same movie title are sourced with (quasi) regularly-staggered phase offsets, customers can perform discontinuous operations by specifying the length of video they want to skip, possibly in integer multiples of the phase-offset duration. Limited continuity in VCR actions can also be provided by the CPE buffer, which can be accessed without removing the customer from the multicast group, and may even allow multicast-group membership changes to occur in a quasi-continuous fashion. However, without adequate mechanisms for using the VoD system resources other than simply sourcing video material in playback mode, continuity in VCR actions can only be provided intermittently at best. In fact, in the multicast VoD systems studied in Chapters 3 and 4, full support for interactive functions requires an individual service, which can only be achieved by dedicating a channel per customer, thereby limiting the scalability achieved by multicast communication. In this chapter, we address how to provide a fully-interactive on-demand service in multicast VoD systems without compromising system scalability and economical viability.

5.1.1 Approach

Our approach in this chapter is twofold. We first propose mechanisms for unrestricted VCR functionality with minimal degradation of system scalability. We then *integrate* both support for VCR actions and the VoD server's batching policy while considering both VoD server scalability and customers' QoS for various realistic scenarios.

First, since full-fledged support for interactive operations requires dedicating "interaction" channels (or I-channels) to execute operations which would otherwise be blocked, *resource reclamation* is needed to prevent the system from degrading to a non-sharing mode. To this effect, we show that the CPE buffer can be actively employed to merge the customer back in synchronization with a "batching" channel (B-channel), by simply prefetching frames or groups of frames while an I-channel is serving the customer in playback mode. Similarly to adaptive piggybacking (cf. Section 4.2.2 of Chapter 4), our proposed framework achieves multicasting "on-the-fly" through decentralized management of non-shared buffers. Thus, both the VoD server and the CPE buffer can work synergistically to decrease the probability of blocking VCR actions while preserving VoD server scalability.

Second, realistic empirical scenarios are needed to identify scalable batching policies for which support for VCR functionality provides good performance, that is, acceptable QoS. In addition to the admission latency and the defection rate due to long waits, customers' QoS in multicast interactive VoD systems now depends on the VCR action blocking probability. Both customers' QoS and VoD server scalability are affected primarily by a combination of four independent factors: (1) the CPE-buffer size and management; (2) customers' request and interaction behavior; (3) the ratio of the number of I-channels to that of B-channels; and (4) the VoD server's batching policy.

Let's briefly examine how these factors interact. First, a larger CPE-buffer size is likely to reduce the load on the pool of I-channels by making a larger portion of the video program available for immediate access. The load on I-channels can also be reduced since a large buffer size makes it more likely for an I-channel customer to find and join a target multicast group after a VCR action. Moreover, active CPE-buffer management can reduce the need for I-channels when interaction behavior is "biased" towards a particular type of VCR action. However, mechanisms to protect intellectual property rights — so that service providers may be able to maintain control of their data and thus stay in business — and affordability constraints will impose an upper limit on the buffer size, which is most likely to be a few minutes' worth of video. Second, customers' reneging behavior before their admission and variations in the request rate impose constraints on the choice of a batching policy and affect VoD server scalability during unpredictable periods of sustained high load. Once customers are admitted, behavioral parameters, such as the level of interactivity or the duration of VCR actions, will affect the performance of the VoD server's support for interactive operations.

Finally, the batching policy adopted by the VoD server and the partition between B- and I- channels also directly affect, in several ways, customers' QoS and VCR functionality. First, for a fixed concurrent channel capacity — usually imposed by the underlying storage capacity and organization (cf. Chapter 2) — the partition between B- and I- channels will determine the tradeoff between the admission latency experienced by new requests, and the blocking probability of VCR actions. Also, for a given number of B-channels, a very effective batching strategy increases the load on I-channels by admitting more customers into the system. Note that the tradeoff between admission and I-channels is less clearcut as it may first appear, since more channels allocated to a frequently-requested movie will increase the likelihood of success in merge attempts from I- to B- channels for that particular movie title, and therefore, will have a positive effect on the availability of I-channels.

The chapter is organized as follows. As a starting point, we first follow through the discussion of Section 3.2.3 of Chapter 3 and present in greater detail how a multicast VoD server can provide support for discontinuous and intermittently-continuous VCR actions by using the CPE buffer. Next, we analyze how, with appropriate I-channel management, unrestricted interactive operations can be incorporated into a multicast VoD system. We then describe the simulation environment used to evaluate VCR functionality. For the sake of realism, we refine the idealized model of customers' interaction behavior introduced in Section 2.6.2 of Chapter 2 to capture several key features, such as frequency and durations of VCR actions, and bias towards "forward" or "backward" interactions. Section 5.4 describes the simulation results on carefully-chosen batching policies under various conditions of customers' behavior and resource allocation. Although we restrict our empirical evaluation to monolithic disk arrays, we discuss the applicability of the proposed framework in clustered storage configurations. Finally, we show how *active CPE-buffer management* can reduce the number of VCR actions when interaction behavior is biased towards a particular type of VCR action. The chapter concludes with Section 5.6.

5.2 Support for Limited VCR Functionality

We present in this section the CPE-buffer support for discontinuous and intermittently-continuous VCR actions, when the VoD server simply sources video material in playback mode.

5.2.1 Interactive Behavior

For clarity, let's briefly review our assumptions on customers' behavior. Customers wait in line until the VoD server decides when to start program transmission based on constraints such as the available capacity of the movie archive on disks or disk arrays. In the most general case, customers may be partially-patient and renege service after waiting a certain amount of time. Such admission behavior determines how customers place their requests on the VoD server. After admission, customers' interactive behavior consists of the T-VoD interactions presented in Section 2.6 of Chapter 2, namely play, resume, stop, pause, abort, fast forward, rewind, fast search, reverse search, and slow motion. As explained in Section 3.2.3 of Chapter 3, the limited support provided *by default* in multicast VoD systems is best understood in terms of the linearity in playback experienced by customers. *Continuous* interactive operations allow a customer to fully control the duration of interaction, which is limited by the amount of video data held in the CPE buffer. *Discontinuous* interactive operations happen in two cases: (1) a customer may suddenly exceed CPE-buffer capacity while performing a continuous action, and (2) a customer may specify the length of video s/he wants to skip (as in fast forward and rewind) in discrete increments predetermined by the phase offsets between adjacent channels carrying the same movie title. In both cases, discontinuous actions are performed by transferring the customer to another multicast group (i.e., by tuning to the appropriate frequency) whose playout point is the closest to that requested by the customer. While "naturally" discontinuous VCR actions such as fast forward and rewind require negotiation of the jump size before the action actually takes place, continuous actions are performed until the viewer either decides to return to playback mode, or until the CPE-buffer capacity is exceeded, whichever happens first.

5.2.2 Continuous Service of VCR Actions

As discussed in Section 3.2.3 of Chapter 3, a small buffer (e.g., 1 – 5 minutes' worth of compressed video data) is likely to be affordable in a near future, especially in comparison with other hardware components required by the CPE (e.g., decompression hardware). The general operation of a CPE buffer during a VCR action is depicted in Figure 5.1. The CPE buffer can be seen as a *sliding window* over the largest *usable* sequential portion of the video (e.g., 7,000 frames, or roughly 4 minutes' worth of video, on the figure). This portion is composed of the video between the most recent frame — the latest frame available for

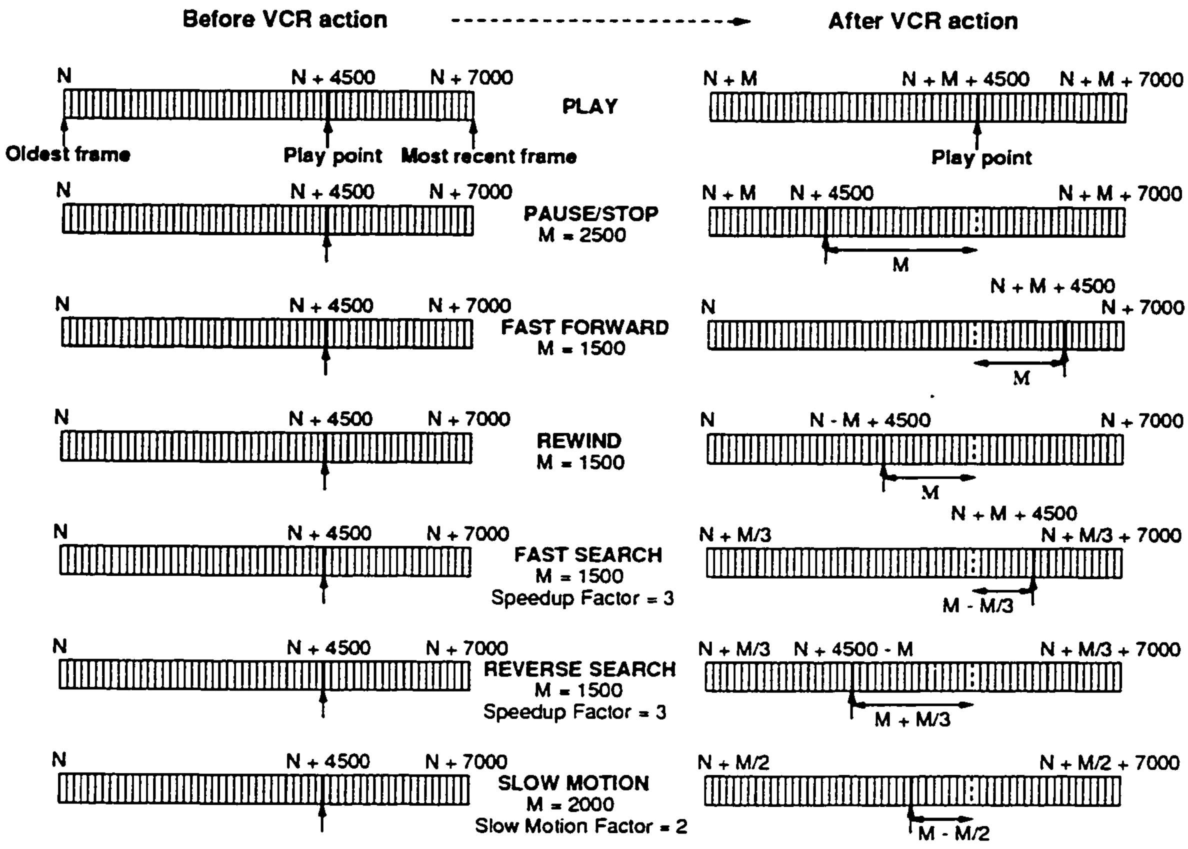


Figure 5.1: Displacement of the play point within the CPE buffer

linear access in the portion of video program currently held by the CPE buffer[1]— and the oldest frame. In between these two frames, the location of the video currently accessed by the customer is known as the *play point*.

Video frames (or more generally group of frames, or video segments) are received at rate R in a synchronous fashion, and those frames already displayed ("past frames") are kept in the buffer, for reverse search and rewind operations. Those prefetched frames that have not been displayed yet are called "future frames." If the CPE buffer is full, the oldest frame is discarded to make room for a recently-received frame. In Figure 5.1 — and throughout the chapter — the positions of the play point and oldest frame are always represented relative to the most recent frame. Thus, each graph can be thought of as a "snapshot" of the usable CPE-buffer content at a given time. Note that this representation of the CPE buffer is a *logical representation* used for the sake of clarity, and it may not necessarily correspond to the actual arrangement of frames on physical storage. We deliberately ignore

[1] The most recent frame usually corresponds to the most recently prefetched frame from the multicast group.

out-of-sequence frames, which are treated as buffer space available for past frames.

Initially, the play point corresponds to the most recent frame and the CPE buffer is progressively filled with past frames as the initial playback continues. Upon execution of VCR actions such as pause, stop, fast reverse or rewind within the CPE buffer, the play point will change as shown in Figure 5.1 while frames are still being received synchronously from the multicast group. Play actions will not change the relative position of the play point with respect to the most recent frame. After a pause or a stop for a duration of M frames, however, the CPE buffer is forced to increase the distance between the play point and the most recent frames by M frames, thus causing a negative displacement of the relative position of the play point. Fast forward and rewind will simply cause a jump of M frames forward or backward. As for fast search, reverse search, and slow motion, the displacement of the relative position of the play point will depend on the speedup factor SP (e.g., 3) and the slow motion factor SM (e.g., 2). For instance, due to the synchronous constraint, a fast search action spanning over M frames will cause an actual displacement of $(1 - \frac{1}{SP})M$. Similar observations can be made for reverse search and slow motion.

Depending on the relative displacement of the play point with respect to the most recent frame, we can now classify VCR actions between "forward" and "backward" interactions. Fast search and fast forward can be considered as forward interactions, since during these interactions, the gap between the play point and the most recent frame is being progressively reduced. All other VCR actions (besides play which leaves the relative position unaffected) are considered as backward interactions. Note that this classification may be counter-intuitive for some VCR actions. In the case of slow motion for instance, the customer is accessing video data that is located forward in the program, but the frames in the CPE buffer are being accessed and displayed at a rate slower than the synchronous rate, hence causing a negative relative displacement of the play point. For the same reason, pause and stop are also considered backward interactions.

5.2.3 Discontinuous Service of VCR Actions

A viewer reaching the most recent or the oldest frame in the CPE buffer while performing a fast search, reverse search, or a slow motion will be forced to resume playback. Consequently, due to its limited size, the CPE buffer can only provide intermittent support for continuous VCR actions. As can be seen from Figure 5.1, discontinuous actions such as fast forward, rewind, and continuous actions such as pause or stop for a short duration, can also be served in a continuous fashion as long as the CPE buffer can store prefetched

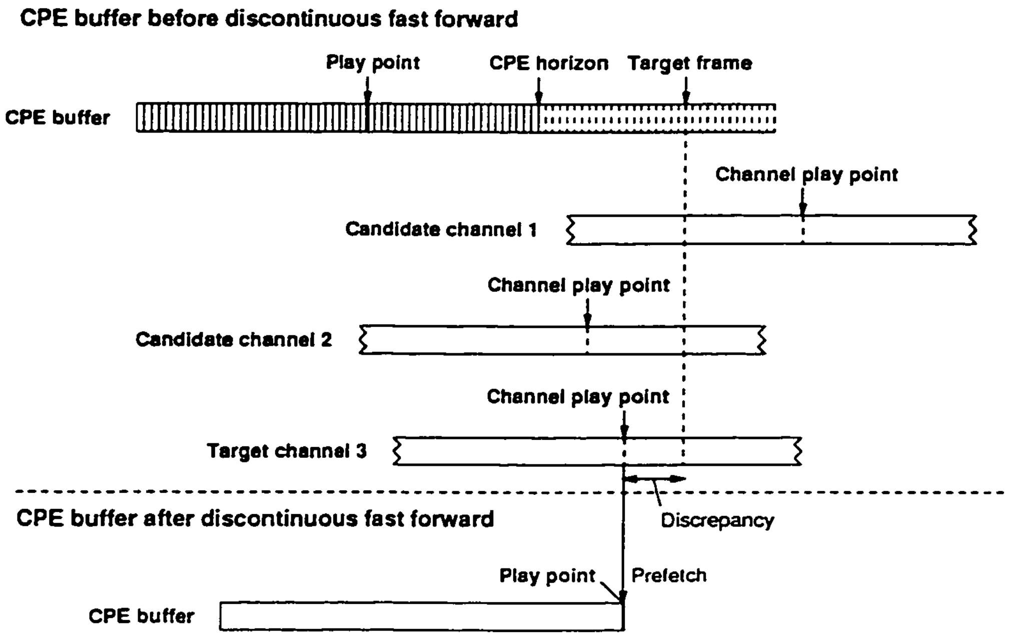


Figure 5.2: Discontinuous fast forward action.

frames from the current multicast group. Otherwise, a multicast group change is required, making the service discontinuous.

In case of a fast forward or rewind request for a video location outside the CPE buffer, the VoD server will first determine the channel whose play point is the closest to the target frame requested by the customer. This operation is illustrated in Figure 5.2, in which the target channel is the candidate whose play point at the time of the request is the closest to the requested target frame. Once a multicast group change has taken place, the entire content of the CPE buffer has to be discarded since the continuity in the frame sequence has been broken.

In case of a pause or stop, multicast group change is needed when the CPE buffer is full and the play point corresponds to the oldest frame. Depending on the CPE-buffer content and on the duration of the interaction, discontinuous pause and stop actions may not necessarily incur discontinuity in the playback seen by the viewer. In the situation represented in Figure 5.3 (case 1), the VoD server sources a "later" candidate channel whose play point has already been prefetched while the customer was in pause or stop state and the CPE buffer was being fed by a B-channel. In this case, group reassignment is seamless and the CPE buffer can discard any frame that has been prefetched after the candidate channel playout point, leaving spare buffer space for past frames. In the situations represented in Figure 5.4 (cases 2 and 3), the target channel is chosen to minimize the jump experienced

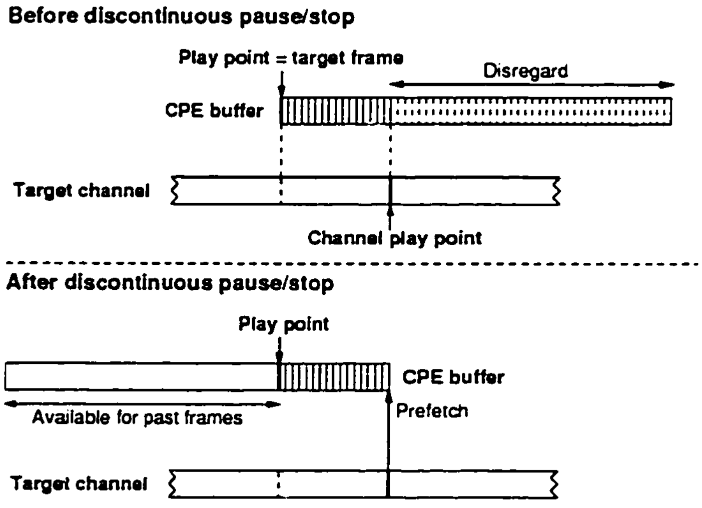

Before discontinuous pause/stop
Play point = target frame
Disregard
CPE buffer
Target channel
Channel play point
After discontinuous pause/stop
Play point
CPE buffer
Available for past frames
Prefetch
Target channel

Figure 5.3: Discontinuous pause/stop: case 1.

when the customer resumes. The CPE-buffer content has to be entirely discarded[2]. Note that the size of the discontinuity is a measure of the QoS experienced by the customer.

If several candidate channels are available in case 1 of a discontinuous pause or stop, the choice of a target channel is arbitrary and depends on the CPE-buffer management. If backward VCR actions are a customer's dominant behavior, one should keep as many past frames as possible in the CPE buffer. In this case, the customer should join the channel with the largest number of discarded prefetched frames (i.e., whose play point is the closest to the customers' play point). In the other cases, one may choose to change the CPE buffer as little as possible and keep as many prefetched frames as possible.

As explained in Section 3.2.3 of Chapter 3, the number of channels allocated to a particular movie title will determine the discontinuity experienced by customers of that movie title in discontinuous VCR actions. If very few channels are allocated to a rarely-requested movie, the likelihood of finding a "nearby" candidate channel following discontinuous actions is very small. In this case, it is safe to assume that group changes will not occur unless expressly requested by customers. As more channels are allocated to the most frequently-requested movie titles, the average phase offset between two adjacent channels can become small enough for the CPE buffer to hold the difference in frames of the two channels, so

[2] In case 3, the CPE buffer may also choose to prefetch frames from the target channel without discarding the frames that have been already prefetched. In this case, the size of the discontinuity is reduced to the offset between the play point of the target channel and the most recent frame held by the CPE buffer. Note that this discontinuity will not be experienced immediately, but instead, when the play point of the CPE buffer reaches the frame corresponding to the play point of the target channel when playback was resumed. Hence, this type of buffer management tends to break the continuity in the frame sequence held by the CPE buffer. A detailed study of this approach, which is more beneficial with large CPE buffers, is left as future work.

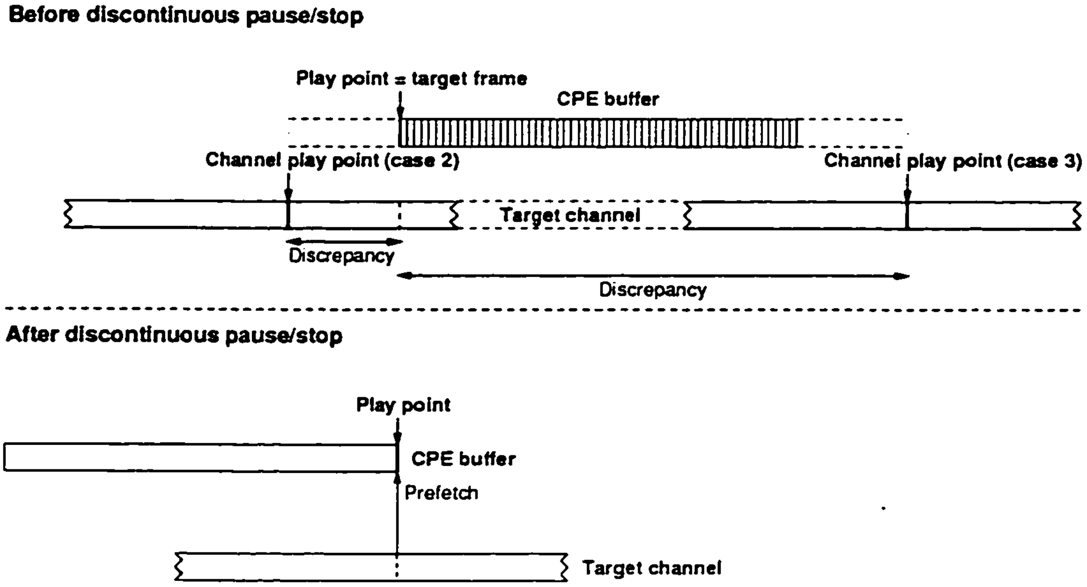

Before discontinuous pause/stop
Play point = target frame
CPE buffer
Channel play point (case 2)
Channel play point (case 3)
Target channel
Discrepancy
Discrepancy
After discontinuous pause/stop
Play point
CPE buffer
Prefetch
Target channel

Figure 5.4: Discontinuous pause/stop: cases 2 and 3.

that a group change is more likely to be performed seamlessly. In any case. whether the phase offset is larger or smaller than the CPE-buffer capacity, the CPE buffer cannot always guarantee a smooth transition between adjacent multicast groups.

5.3 VoD-Server Support for VCR Actions

Interaction functionality in VoD servers dedicating their resources to sourcing video materials in playback mode is, at best, limited to discontinuous and intermittently continuous VCR actions. This fact led researchers [64, 12, 31] to recognize that providing full-fledged support for interactive operations requires dedicating a subset of the concurrent channel capacity to customers interactions. A community access TV cable tree (CATV-Tree) is a good candidate since its total bandwidth can be partitioned into channels for TV broadcast and channels for interactive multimedia [74]. A modular architecture combining both a video multiplexer and a video switch to accommodate such partitioning is presented in [62]. Used in conjunction with partial caching of programs, I-channels are allocated on-demand to handle interactive operations which would otherwise be blocked or served in a discontinuous manner. In this section, we show that with an appropriate management, both the VoD server and the CPE buffer can work in synergy to make sharing and scalability transparent to the users, while providing a service closer to true interactivity. Our proposed framework is based on the observation that the low-latency CPE buffer can be actively employed to merge the customer back in synchronization with a B-channel, by simply prefetching frames or groups of frames, once the customer resumes playback while being served by an I-channel.

5.3.1 Related Work

VoD systems combining request batching and VCR functionality were initially presented in [12], [31], and [64]. In [12] and [31], support for continuous service of pause operations was simulated for an NVoD server but merge operations from I- to B- channels were either ignored ([31]) or did not guarantee continuity in video playout ([12]). In the *split-and-merge (SAM) protocol* [64], all interaction types introduced in Section 2.6 of Chapter 2 are served by allocating I-channels as soon as a VCR action request is issued. When playout resumes, the VoD server attempts to merge the user back into a B-channel, using a portion of the *synch buffer*, which is a buffer located at access nodes and shared by all the users. The maximum buffer size dynamically allocated to a user is arbitrarily set by the service provider. Should a merge attempt fail, a request for a new B-channel is then initiated. Although the SAM protocol, and to a lesser extent the schemes presented in [12] and [31], dramatically increase the capacity of the system in comparison with T-VoD systems which do not use batching, these schemes lack scalability in both admitting newly-arrived requests and servicing interactive operations.

In the SAM protocol, B-channels are allocated in two possible ways. First, upon their arrival, customers' requests for a particular movie are forced to wait for an arbitrary amount of time so the batching factor for that movie may be increased. This batching policy is known as *forced-wait* [11]. Second, B-channels are also started *on-demand* at a random location in the video, for customers who could not be merged back into an existing multicast group after their VCR action was served by an I-channel. Both forced-wait and on-demand batching policies are non-scalable "greedy policies" which, as seen in Chapter 4, cause uncontrolled customers' QoS degradation, following a sudden increase in traffic intensity and in customers' level of interactivity. In the latter case for instance, consecutive pause actions in SAM tend to reduce the amount of synch buffer available to the customer, thereby increasing the number of requests for B-channels. Then, these B-channels, allocated on-demand, will not be available for admission of new incoming requests and the overall throughput of the VoD server may be significantly reduced. To further illustrate the scalability problem, B-channels in [31] are allocated as in NVoD for customers' admission, and on-demand when admitted customers resume from pause actions. Even in this "partially-scalable" situation, it is shown in [31] that given assumptions on customers' behavior, the required channel capacity ensuring a predetermined latency in both admission and pause actions, and a minimum defection rate, grows *linearly* in request arrival rate. Clearly, allocation of B-channels has to be isolated from customers' interaction behavior. Also, an "admission queue" has to be implemented to batch the incoming requests in a scalable fashion as presented in Chapter 4 and for throughput guarantees.

Next, the shared synch buffer, statically allocated by the VoD server and located at the access nodes, is used exclusively for merge operations[3]. As documented in [64], customers only use, on the average, a small fraction of their dynamically allocated buffer, and the resulting buffer utilization is low. Had the synch-buffer capacity been used similarly to a CPE buffer as in Section 5.2.2, more interactions would have been served without I-channel allocation, hence reducing the load on I-channels. In addition, the time needed to merge customers back into a multicast group may be considerably reduced, or even eliminated, if buffer space is used to keep frames that have already been prefetched from an I-channel. This situation, elaborated on in Section 5.3.2, can happen frequently after pause, stop, or reverse search actions. A shortened merging time clearly results in a decreased I-channel holding time and a subsequent higher availability of I-channels to serve other VCR actions. In contrast, longer resource reclamation in SAM leads to a blocking probability and delay in VCR actions that grow linearly in traffic intensity[4]. Furthermore, in the case of nonstationary arrival rate of requests, a static synch-buffer capacity will result in successive periods of wasted buffer space due to poor buffer utilization, and of reduced per-customer buffer space due to load increases. In the latter case, a smaller available per-customer synch buffer will cause excessive I-channel holding time and uncontrolled loss of QoS in VCR actions.

Once customers have been merged back into a multicast channel, additional bandwidth is needed in SAM to deliver the video data from the synch buffer to the CPE. If this scheme is implemented in a bandwidth-limited CATV network, since all CPEs already receive all multicast channels, a recently-merged customer will be delivered the same video data twice: once via a B-channel, and once from the synch buffer by using the bandwidth available to I-channels (the phase offset between both transmissions corresponding to the merging time). Unlike the shared-buffer approach presented in [64] and in Section 4.2.1 of Chapter 4, no additional bandwidth is needed if CPE buffers, provided by the customers themselves, are

[3] Note that this contrasts with the costly caching approach presented in Section 4.2.1 of Chapter 4, in which the VoD server memory is used to retain moving portions of certain videos as they play.

[4] Similarly, without CPE-buffer support for resource reclamation as in [31], the required channel capacity ensuring a predetermined latency in pause actions grows linearly in request arrival rate.

used for I-channel reclamation as presented hereafter, in Section 5.3.2.2. In addition, we will show that active use of the CPE buffer achieves a higher QoS and scalability in VCR actions. These observations are quite important as a shared buffer may seem a cheaper alternative (since the likelihood of all viewers using the synch buffer at the same time is usually pretty low, thus reducing buffer cost). Although more work is needed to determine which of the two achieves better cost-effectiveness in a large-scale deployment context, it should be noted that both schemes are not mutually exclusive. For instance, a VoD server could allocate a synch buffer for bufferless clients, while customers with a CPE buffer would use our scheme. In any case, we argue in this chapter that decentralized management of non-shared buffers should not be dismissed in favor of the shared-buffer approach for the following additional reasons: (1) unlike in the shared-buffer case, no extra buffer, bandwidth, network management, and processing power are needed as the service provider expands or incurred during resource reclamation; in the SAM case, managing a large centralized or distributed synch buffer and determining a suitable maximum individual buffer size are further complicated as the VoD service is accessed by an increasingly large number of customers, who may be generating time-dependent request arrival patterns; (2) buffer costs are divided among customers; and (3) as we saw in Section 5.2, a reasonably-sized CPE buffer is likely to be affordable in a near future. In fact, points (2) and (3) work hand in hand to drive the cost of CPEs down.

Lastly, when no resources are available to handle continuous service of interactive operations due to a heavy load of requests, it is critical for the service provider to offer discontinuous service as explained in Section 5.2. In the SAM protocol, blocked interaction requests are queued, and, therefore, experience unpredictable delays which limit the "on-demand" nature of the service.

5.3.2 Framework for Scalable Interactions

5.3.2.1 I-Channel Allocation

I-channels are allocated when a customer exceeds its CPE-buffer limit in three different cases: (1) the play point reaches the oldest buffered frame during a stop, pause, reverse search or a slow motion action; (2) the play point reaches the most recent frame during a fast search action; (3) a fast forward or rewind action is requested for a frame located outside the CPE buffer. This is summarized in the left side of Figure 5.5. Note that even though pause and stop actions do not require data delivery, an I-channel has to be allocated as soon as the CPE buffer is exceeded in order to ensure continuous playback when the customer decides to resume. If no I-channel is available, fast search, reverse search and slow motion actions

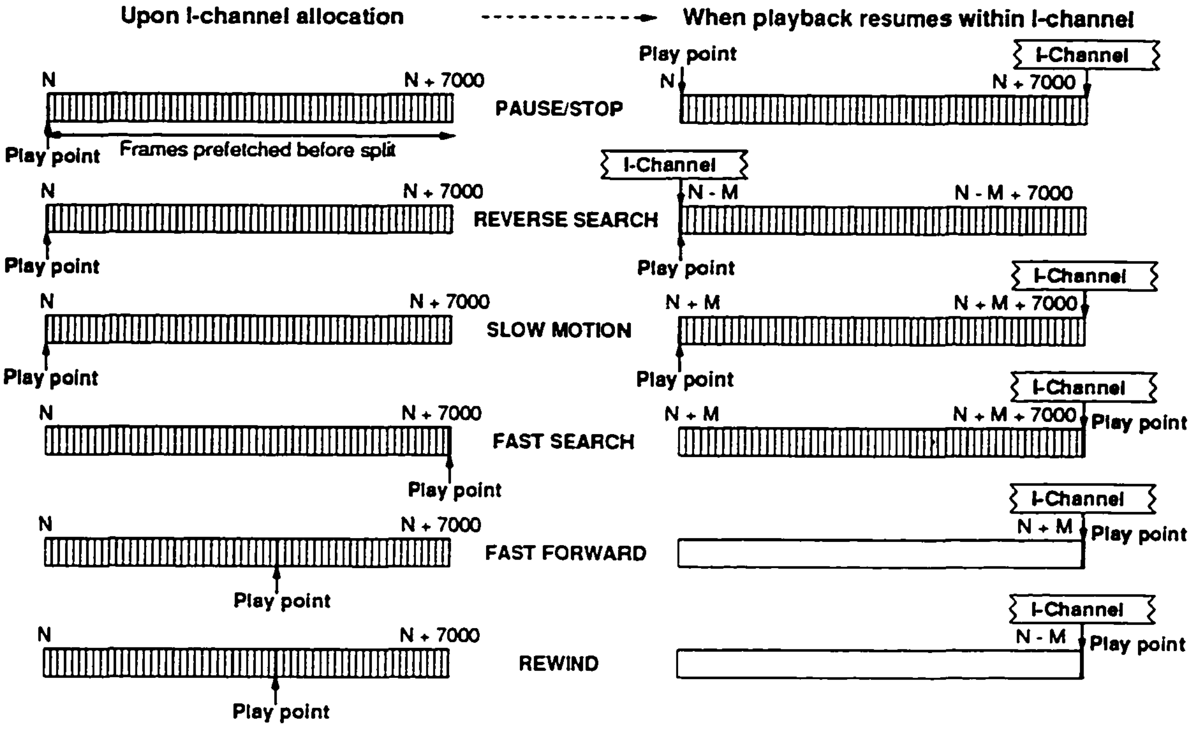


Figure 5.5: CPE buffer before I-channel allocation and after playout resumes.

will be blocked, whereas in other cases, the VCR action will be served in a discontinuous fashion as explained in Section 5.2.3. In summary, VCR actions can be either served in a continuous fashion by the CPE buffer or an I-channel, blocked, or served in a discontinuous fashion.

If an I-channel is readily available, the customer is served in interaction mode, which may require the VoD server to adopt a specific retrieval and delivery policy when the data delivery rate is altered (e.g., during a slow motion, fast search or reverse search), as we described in Section 2.6.1 of Chapter 2. Similarly to our study of T-VoD in Section 2.6 of Chapter 2, we assume, for simplicity, that the linear frame sequence is preserved during fast or reverse search actions. This constraint is met if frames are transmitted at n times the normal delivery rate [35] or if fast access is performed by retrieving a version or a component of the movie intended for that purpose [53, 70, 88]. As for slow motion, we simply assume that the most recent frame is prefetched and the oldest frame discarded at a slower rate, without changing the relative position of the play point.

The right side of Figure 5.5 represents the CPE buffer when playback is resumed within an I-channel. In the case of a fast forward and rewind action, all frames in the CPE buffer have to be discarded since the linear sequence is broken. Note that although fast forward

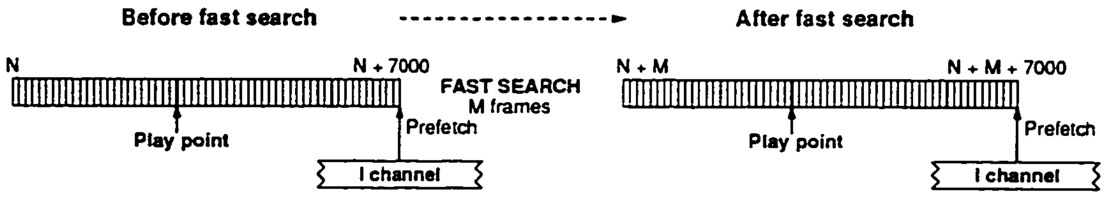


Figure 5.6: Fast search performed in an I-channel

and rewind are discontinuous actions, they are served in a continuous fashion if an I-channel is available to accommodate the displacement in play point. A fast search action should be executed by fetching frames from the I-channel at a rate $SP \times R$ as represented in Figure 5.6, without changing the relative position of the play point and the number of future frames within the CPE buffer. In the case of a slow motion, pause or stop action, we assume that the relative position of the play point remains unchanged. Thus, during a slow motion action, the CPE buffer is actually prefetching the most recent frame while the customer is displaying the frame located in the play point, which corresponds to the oldest frame. The operation of an I-channel during a reverse search is more complex. During the interaction itself, frames are prefetched in reverse order, at a faster rate. Thus, the CPE buffer will stay full and the play point will not be changed. Note that this is the only situation where the most recent frame does not correspond to the most recently prefetched frame. When playout resumes, we assume that the I-channel will start sending the frame corresponding to the most recent frame, so that, similarly to slow motion, pause and stop actions, the play point is unchanged and the entire content of the CPE buffer is kept as future frames. The reason for keeping as many future frames as possible is to shorten the duration of merge operations, as we shall see in the next section.

5.3.2.2 Merge Operation

In order to prevent the system from degrading to a non-sharing mode, a key feature of the proposed scheme is that the VoD server attempts to merge customers back to an ongoing B- or I- channel in play mode as soon as they resume playback within an I-channel. A merge operation is basically to search for a *target channel*, which is an active channel in play mode carrying the same movie title as the I-channel to be merged, and whose play point temporal location in the video program is ahead of the customers' play point but no more than d_{CPE} ahead, where d_{CPE} is the maximum duration of video held by the CPE buffer and played at playback rate. The last condition is needed to ensure the availability of enough CPE-buffer space to prefetch frames from the target channel while the merge

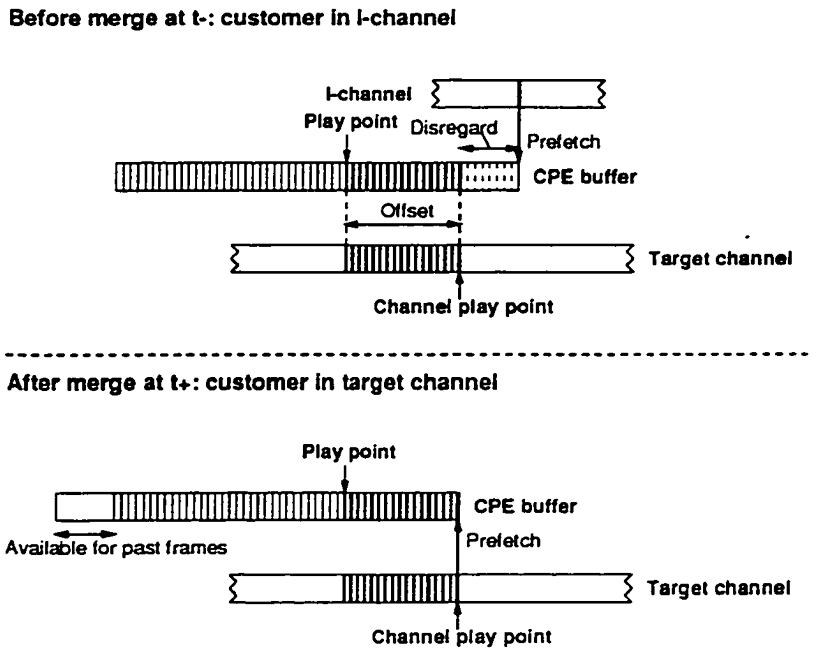


Figure 5.7: Instantaneous merge operation.

operation is taking place.

Depending on the relative position of the customers' play point within the CPE buffer right after the VCR action — indicated in the right side of Figure 5.5 — we can distinguish two situations. First, as illustrated in Figure 5.7, the play point of the target channel may already reside in the CPE buffer. This is possible, for instance, after a pause, stop, slow motion or reverse search performed in an I-channel, in which case the CPE buffer is full and the play point corresponds to the oldest frame in the CPE buffer. In this situation, the play point of the target channel, if any, is already present in the CPE buffer and the merge operation can then be performed *instantaneously*. This is illustrated in Figure 5.7[5]. Once the customer has been merged into the target channel, two management choices are available to the CPE buffer. First, the CPE buffer can opt to discard all frames that are ahead of the play point of the target channel, so the most recent frame may be in synchronization with the target channel. Second, it is also possible for the CPE buffer to discard the frames sent by the target channel that are already present in the CPE buffer, while moving forward the relative position of the play point. In both cases, the buffer space left by discarding frames becomes available for past frames. If several channels are found eligible for a merge operation, the choice of a target channel is purely arbitrary and depends on the CPE-buffer management. If the play point of the candidate channel is close to the customers' play

[5] Note that the merge operation in this case is similar to a discontinuous pause/stop, case 1 in Section 5.2.3

point, a large portion of prefetched frames will have to be discarded, thus making more space available for past frames. Choosing such a target channel would be better suited for interactive access patterns dominated by backward interactions. If the play point of the target channel is close to the most recent frame held by the CPE buffer, most prefetched frames would be kept after the merge. This choice, assumed by default, allows to save bandwidth by minimizing the number of prefetched frames that would otherwise have to be transmitted twice.

The second merge situation occurs when the target channel's play point has not yet been prefetched, for instance, after fast search, fast forward, or rewind actions, which move the relative position of the customer's play point within the CPE buffer to the most recent frame. Since a few future frames are held by the CPE buffer, the duration of the merge operation corresponds to the offset between the play point of the candidate channel and the most recent frame currently held by the CPE buffer. This is illustrated in Figure 5.8.

A merge operation comprises 3 steps: (1) free sufficient CPE-buffer space by discarding past frames; (2) prefetch frames from the target channel while displaying frames received from the I-channel, which is held for the whole duration of the merge operation; (3) after a duration corresponding to the offset between the respective play points, merge into the target channel and release the I-channel. Clearly, an I-channel that is in the process of being released cannot be eligible as a target channel for another active I-channel. If several channels are found eligible, the target channel corresponds to the minimum offset between play points. If the customer initiates a VCR action while a merge operation is taking place, the frames that have already been prefetched from the target channel are discarded and the merge operation is aborted. Note that in this case, some past frames were unnecessarily discarded due to unpredictable customer behavior.

A merge operation may fail for several reasons. First, it is not possible to find any candidate channel, in which case playout will continue by fetching frames at playback rate from the I-channel until the next VCR action. Second, a customer may interrupt a merge operation by initiating a VCR action. In both cases, consecutive VCR actions will change the relative position of the play point within the CPE buffer, which may no longer correspond to the oldest or to the most recent frame when the next merge attempt takes place. In both situations, while the customer is being served by an I-channel, the CPE-buffer management should strive to keep as many prefetched frames as possible in order to shorten the duration of a merge operation. This is achieved by using CPE buffer as indicated in Figure 5.1 when the customer is in playback mode, and as indicated in Section 5.3.2 during VCR actions.

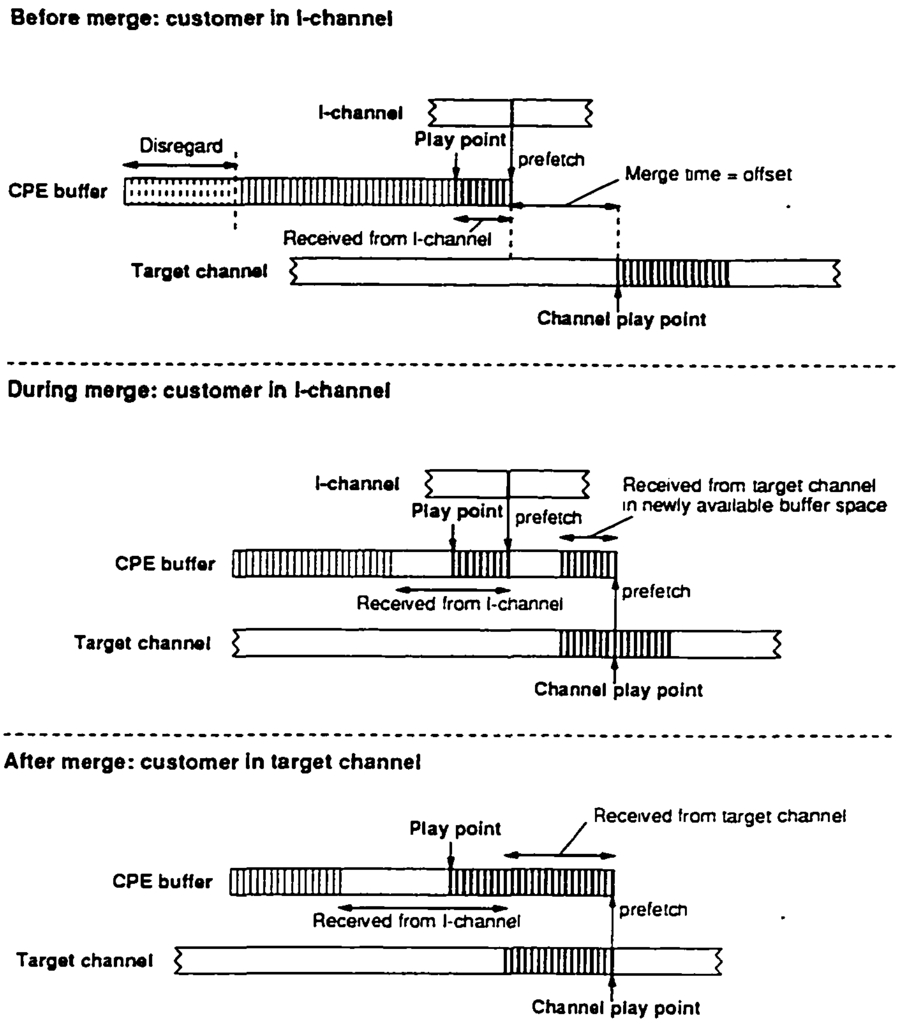


Figure 5.8: General merge operation.

In both merge and discontinuous operations, on-going B-channels will be scanned first for eligibility, and if no candidate B-channel has been found, I-channels in play mode are next to be examined. If the target channel is an I-channel, it is changed to a B-channel after the merge operation: such a B-channel is called a *pseudo-B-channel* as it is not used for admitting newly-arrived requests and it originates from the pool of I-channels. Likewise, if customers leave a pseudo-B-channel after issuing a VCR action, and only one customer is left in the channel, the pseudo-B-channel is reverted back to an I-channel and will be merged as soon as possible. Finally, when all customers depart from a regular B-channel used initially for admission, we will simply assume that this particular B-channel will be kept active to provide support for future merge and discontinuous operations.

5.4 Numerical Results

5.4.1 The Simulation Environment

In our simulations, we considered NVoD[6], QVoD-enhanced NVoD. and QVoD-h batching policies introduced in Chapters 3 and 4. We selected these policies for their scalability and support for discontinuous actions. (Note that in QVoD-enhanced NVoD discontinuous VCR actions are then only provided *in an average sense.*) The metrics of interest used to evaluate the various above-mentioned batching policies are the average latency experienced by customers in receiving the requested service. denoted by $\overline{EW}$, and the average defection rate L_r. For each metric, we also consider the dispersions D_{EW} (D_{L_r}), defined as the coefficient of variation of the waiting times (defection rate) for each movie title EW_m (L_{rm}). Lower values of D_{EW} (resp. D_{L_r}) indicate more homogeneous allocation policies which lessen variability in customers' waiting times (defection rates).

In the rest of this chapter, we assume that customers arrive according to an aggregated Poisson process with parameter λ, and request $N = 10$ movie titles of length $L = 100$ minutes each, stored on a monolithic disk array with capacity $s = 200$ channels. The traffic intensity is $\rho = \frac{\lambda L}{s}$ and movie popularities are given by Zipf's law. In case of partial patience, we use the reneging behavior model introduced in Section 3.3.1 of Chapter 3. As in Section 3.4.2 of Chapter 3, we define a *patience factor* as $\beta = \frac{\overline{\tau}}{L}$ to distinguish between *impatient* (e.g., $\beta = 0.01$) and infinitely-patient customers ($\beta = \infty$).

Once admitted, customers interact with the VoD server according to a customer activity model. We further refined the model presented in Section 2.6.2 of Chapter 2, to include the *bias* of interaction behavior, which indicates whether interactive operations are dominated by backward or forward actions, or evenly distributed between the two. The new customer activity model is depicted in Figure 5.9. Similarly to the model of Figure 2.14, durations $d_i, i = 0 \cdots 8$ and transition probabilities $P_i, i = 0 \cdots 9$ are assigned to a set of states corresponding to the different VCR actions. These measurements can be collected as shown in Figure 2.15. Each viewer stays in each state for an exponentially-distributed period of time, unless the beginning or the end of a movie is reached (the customer exits the system normally in the latter case). Also, several key parameters are captured by this representation of viewers' activity. First, the level of interactivity can be adjusted by assigning higher or lower transition probabilities from the play/resume state to other states. Next, the duration of VCR actions is included in the model. Finally, the bias of interaction behavior can be

[6] We shall henceforth call the non-work-conserving system "NVoD" and call the work-conserving one "NVoD-VCR."

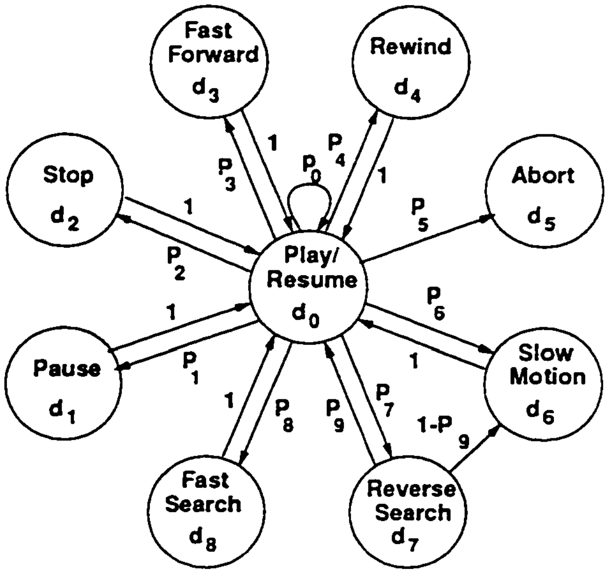


Figure 5.9: Transition diagram for the customer activity model.

set by tuning transition probabilities from play/resume in favor of backward or forward actions. Note that transitions between interactive states such as slow motion and reverse search are also modeled.

For tractability, we defined a set of 12 distinct interaction behaviors. These behavior types extend those considered in Section 2.6.3 of Chapter 2. First, with the exception of the duration in play/resume state being fixed at 10 minutes and in abort state (of null duration), VCR actions are assumed to be of equal length and can be either *long* (5 minutes) or *short* (1 minute). As represented in Figure 5.9, the transition probability from any other state than play/resume to play resume is 1, with the exception of P_9 set to 0.5. We also assume that slow motion is requested only after a reverse search (i.e., $P_6 = 0$). For other transitions, customers can be either *very interactive* (VI) or *not very interactive* (NVI), and bias in VCR actions can be either *forward* (FB), *backward* (BB), or *neutral* (NB). Transition probabilities are summarized in Table 5.1. Note that by setting equal durations, we can control bias and interaction intensity by simply changing the transition probabilities. It is also possible, as was done in [64], to assign the same value to all transitions from play/resume to another interactive mode. In this case, one can control bias and interaction intensity by assigning different durations to different VCR actions.

Support for VCR actions provided to customers in B-channels is evaluated by measuring the relative fractions of VCR actions which are (1) blocked; (2) served by the CPE buffer (CPE-actions); (3) served by an I-channel (I-actions); and (4) served in a discontinuous

Behavior	*Play/Res.*	*Abort*	*Stop*	*Pause*	*RW*	*FF*	*RS*	*FS*
NVI,NB	0.75	0.02	0.03	0.04	0.04	0.04	0.04	0.04
NVI,BB	0.75	0.02	0.03	0.04	0.07	0.01	0.07	0.01
NVI,FB	0.75	0.02	0.01	0.01	0.01	0.09	0.01	0.1
VI,NB	0.5	0.04	0.06	0.08	0.08	0.08	0.08	0.08
VI,BB	0.5	0.04	0.06	0.08	0.14	0.02	0.14	0.02
VI,FB	0.5	0.04	0.02	0.02	0.02	0.19	0.02	0.19

Table 5.1: Transition probabilities from play/resume.

fashion[7]. This classification allows us to evaluate both customers' QoS in service received, as measured by their ability to obtain continuous service (fractions of I- and CPE- actions), and customers' *QoS degradation*, which can be either graceful (fraction of discontinuous actions) or not (fraction of blocked actions). Discontinuous actions are always pause, stop, fast forward, or rewind actions for which no I-channel was found to accommodate seamless playback resumption. Blocked actions are always fast search, reverse search, or slow motion actions which were blocked once the CPE-buffer capacity is exceeded.

5.4.2 Infinitely-Patient Customers

5.4.2.1 Choice of a Batching Policy

As a first step, we performed a set of simulations to assess VCR functionality in the case of infinitely-patient customers. We assume short interaction durations (1 minute), low interactivity, neutral bias, and accordingly, a CPE-buffer size set to 1 minute. Our methodology compares performance in batching, discontinuous operations, and continuous operations. Before evaluating the performance of the proposed scheme for VCR capability, we assume that a VoD server has to choose a scalable batching policy among NVoD, NVoD-VCR, QVoD-enhanced NVoD, and QVoD-h (simply denoted by QVoD in the remainder of this chapter) with MFQL scheduling.

This is illustrated in the first graph of Figure 5.10, which represents the average latency experienced by customers before admission in the four batching policies, and in two sets of partition between B- and I- channels, $(B, I) = (200, 0)$ and $(150, 50)$. In NVoD, NVoD-

[7] We will henceforth simply refer to these VCR actions as discontinuous actions, regardless of whether these actions are, by nature, discontinuous (fast forward, rewind), or continuous (pause, stop).

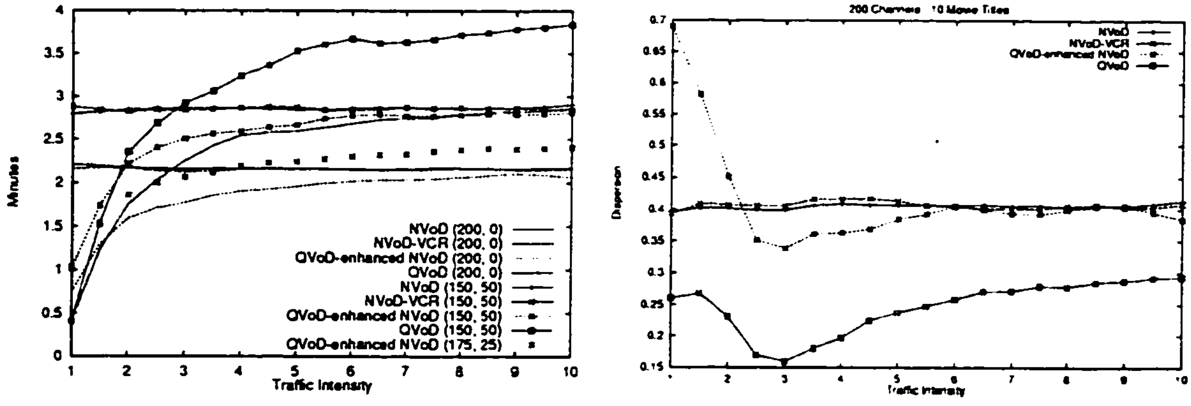


Figure 5.10: Average and dispersion of admission latency for infinitely-patient customers.

VCR, and QVoD-enhanced NVoD, the partition of the B-channel capacity is chosen such that customers' average latency is minimized. Each configuration reaches a plateau pretty quickly as the traffic intensity increases, indicating scalability. For low traffic intensities. NVoD and NVoD-VCR force customers to wait unnecessarily for half a phase offset and both QVoD-enhanced NVoD and QVoD provide a quicker response time. As the traffic intensity increases, QVoD-enhanced NVoD appears to yield the lowest latency. QVoD performs the worst, as can be seen from the fact that the service latency achieved with 200 B-channels is comparable to that achieved by NVoD, NVoD-VCR and QVoD-enhanced NVoD with 150 B-channels only. The corresponding dispersion of the admission latency (depicted in the second graph of Figure 5.10) indicates that, with the exception of low traffic intensities, the improvement provided by QVoD-enhanced NVoD over NVoD and NVoD-VCR does not yield greater unfairness among movie titles. Finally, QVoD yields the fairest allocation, at the cost of increased latency.

We measured the performance in discontinuous operations by focusing on the partition $(B, I) = (200, 0)$. As shown in Figure 5.11, the average fraction of blocked VCR actions is similar for all systems. Thus, the cost of providing work-conserving service in NVoD-VCR does not seem justified when NVoD and QVoD can provide comparable results. Flat curves confirm the fact that discontinuous VCR functionality is, by essence, a scalable service in multicast VoD systems.

The slightly better performance of QVoD can be explained by considering the average discrepancy of discontinuous VCR actions (Figure 5.12), which measures the relative difference between the displacement initially requested by a customer (and eventually blocked) and the actual displacement that occurred when that customer was served in a discontinu-

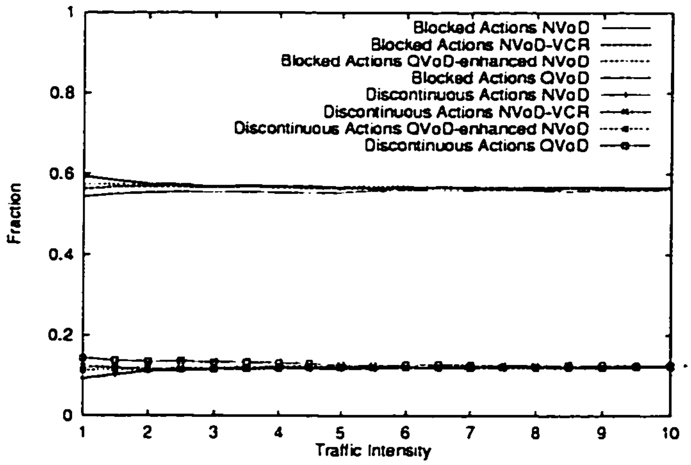


Figure 5.11: Average fraction of blocked and discontinuous VCR actions.

ous fashion. Discontinuous operations are, in general, backward actions (pause, stop, and rewind), which extend the subjective duration of programs. As can be seen in Figure 5.12, the average relative discrepancy experienced by customers in discontinuous actions is reduced in the QVoD case since more channels to more frequently-requested movies. A lower discrepancy in QVoD will thus tend to reduce the extension in program subjective duration due to backward-dominated discontinuous actions. Since the absolute values of the average duration in pause and stop actions, and of the average jump sizes in fast forward and rewind actions are unchanged, the relative fraction of discontinuous actions will then be greater in the QVoD case, as illustrated in Figure 5.11. The fraction of CPE-actions (31%), which depends mainly on the relationship between CPE-buffer size and interaction durations, being unchanged, the fraction of blocked VCR actions will finally be lower in the QVoD case. Note that the fractions of blocked and discontinuous actions (and similarly CPE- and I-actions) presented here and in the remainder of this chapter, are usually constant and vastly independent of the traffic intensity. This confirms that our proposed framework for VCR functionality is, in essence, a scalable service in multicast VoD systems.

An additional observation can be made from the average discrepancy of discontinuous VCR actions (Figure 5.12), which measures the difference between the displacement initially requested by a customer (and eventually blocked) and the actual displacement that occurred when that customer was served in a discontinuous fashion. It is shown in Chapter 3 that QVoD-enhanced NVoD incurs a larger dispersion in the average phase offset among all movie titles as customers' patience increases. This observation led us to question the applicability of QVoD-enhanced NVoD when discontinuous VCR functionality, important to customers'

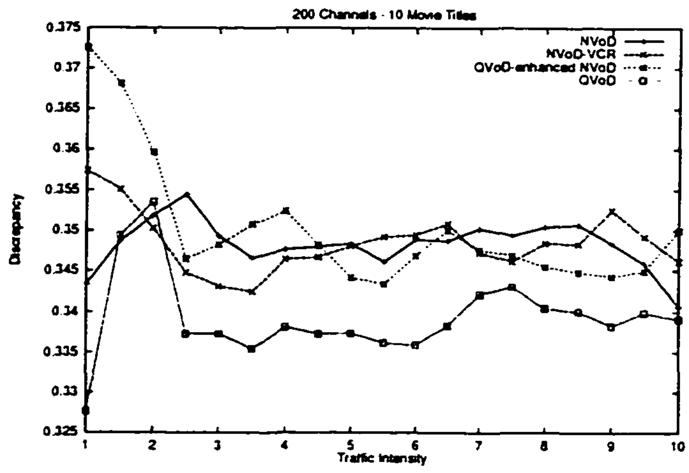


Figure 5.12: Average discrepancy of discontinuous VCR actions.

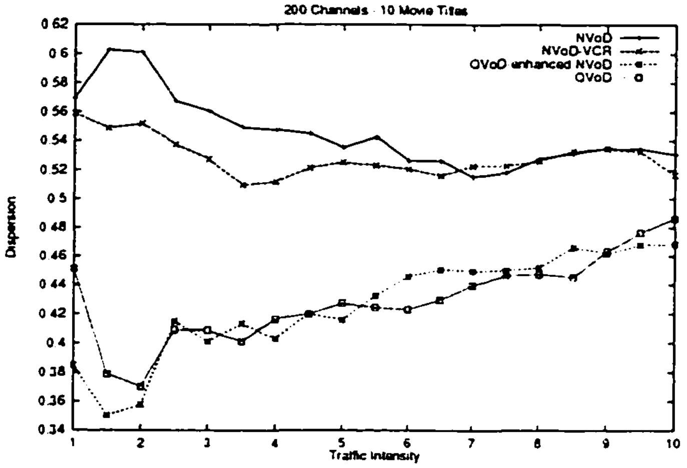


Figure 5.13: Dispersion of average fraction of blocked VCR actions.

QoS, is only provided in an average sense. Nonetheless, Figure 5.12 indicates that, for our particular model of interaction behavior, customers experience the same discrepancy in QVoD-enhanced NVoD. We also found that the dispersion in discrepancy calculated over all movie titles was lower in the QVoD-enhanced NVoD case. Lastly, a greater dispersion in the average fraction of blocked actions in NVoD and NVoD-VCR (Figure 5.13) indicates that keeping a constant phase offset for each movie title is not well suited for exponentially-distributed interaction durations.

Support for continuous VCR actions is evaluated next by considering the partitions $(B, I) = (175, 25)$ and $(B, I) = (150, 50)$. Intuitively, and as confirmed by our simulation results, by allocating more B-channels to frequently-requested movies, the QVoD batching policy increases the probability of merge success, while reducing the fraction of successful

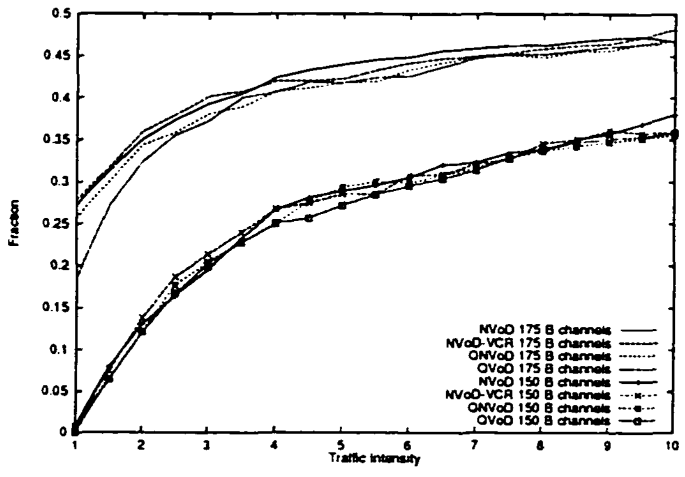


Figure 5.14: Average fraction of blocked VCR actions.

merges to I-channels. Nonetheless, Figure 5.14 shows that all batching policies perform comparably as the number of I-channels is increased. Similar results were obtained for the corresponding dispersion, for the number of discontinuous VCR actions and I-actions, whose dispersion decreases as the number of I-channels increases, since I-channels are completely shared, thus favoring customers of most popular movies. In addition, the observed discrepancy in the fraction of discontinuous actions between QVoD and other batching policies has little impact on the overall blocking probability since the fraction of discontinuous actions decreases. Thus, when a batching policy is to be selected, priority should be given to the latency experienced by customers before service. QVoD-enhanced NVoD is then the best candidate for infinitely-patient customers. Note that even though NVoD systems naturally provide support for limited and scalable VCR capability, such advantage is limited when continuous interactive operations are supported.

As the traffic intensity increases, the I-channel capacity becomes slowly saturated, resulting in a decrease in the *I-channel holding time*, as illustrated in Figure 5.15, which indicates the amount of time that an I-channel is being held by a customer, from the beginning of the first I-action until the I-channel is released, following a successful merge operation[8]. This is because the pool of I-channels is completely shared among customers. Thus, as the traffic intensity increases, I-channels are progressively allocated to customers viewing the most-frequently requested movies, for which the probability of merge success is greater than that of customers viewing other movies. This trend was confirmed by the corresponding dispersion in merge success. Figure 5.15 also shows a greater I-channel holding time for

[8] We assume that the I-channel holding time also includes the time spent as a pseudo-B-channel, if applicable.

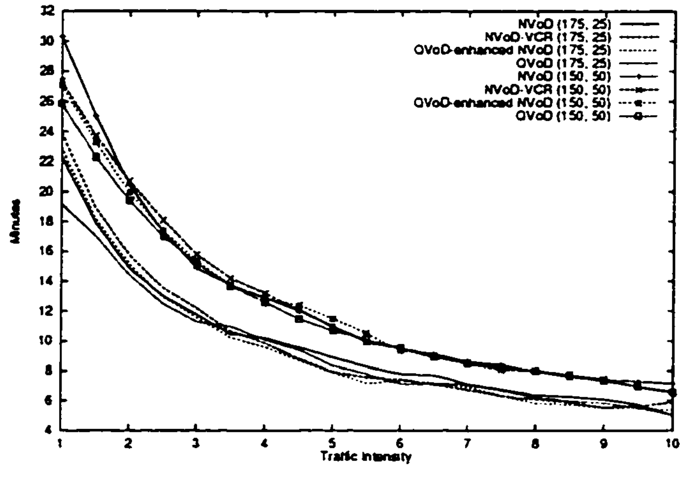


Figure 5.15: Average I-channel holding time.

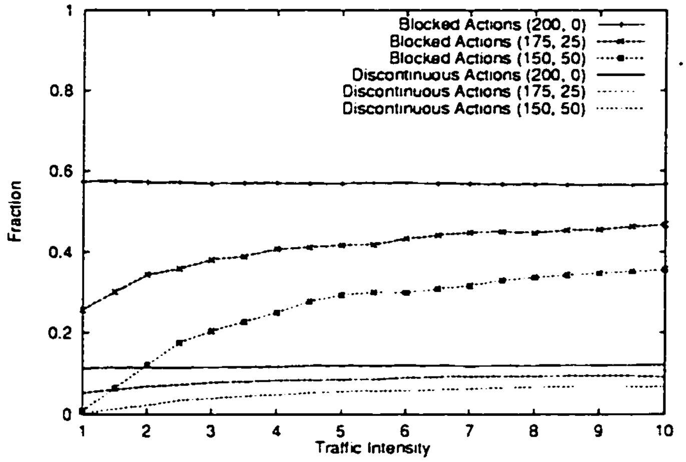


Figure 5.16: Average fraction of blocked and discontinuous actions for QVoD-enhanced NVoD.

the (150, 50) partition. This corresponds to the large availability of I-channels for merge operations, confirmed by a greater fraction of successful merge operations to I-channels in this case.

5.4.2.2 Choice of a Partition between B- and I- Channels

If more B-channels are allocated, a lower admission latency can be achieved at the cost of a higher VCR actions blocking probability. We investigated this tradeoff by simulating the various batching policies in three different partitions, $(B, I) = (200, 0), (175, 25)$, and $(150, 50)$. Figures 5.16, and 5.17 represent the breakdown of VCR actions in QVoD-

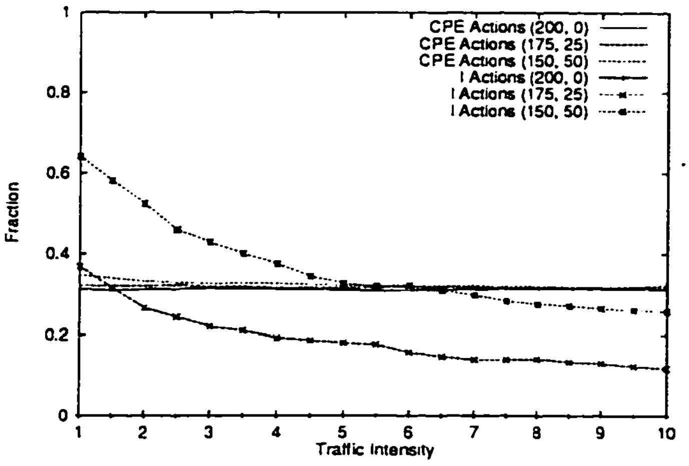


Figure 5.17: Average fraction of CPE- and I- actions for QVoD-enhanced NVoD.

enhanced NVoD. The fraction of blocked and discontinuous actions (58% for $(B, I) = (200, 0)$) can be reduced by half if the number of I-channels is increased from 0 to 50, for an increase of 1 minute (60%) in admission latency (cf. first graph of Figure 5.10).

We are thus assured that, for a fixed channel capacity, the choice of a partition between B- and I- channels will depend on which aspect of customer service is valued most, admission latency or QoS in VCR actions, and on the relative fraction of CPE-actions. If, for instance, the CPE buffer is small, few VCR actions will be served by the CPE buffer and priority should be given to reserving enough I-channels. In each figure, however, the quasi-parallelism that can be observed, for the same batching policy, among curves corresponding to different partitions, indicates *almost no loss of scalability caused by changes in the breakdown between B- and I- channels*.

5.4.2.3 Comparison with the SAM Protocol

Figure 5.18 depicts a comparison between our proposed scheme and the SAM protocol for an arbitrary batching policy (NVoD in our case). The SAM protocol was simulated by using the CPE buffer only for non-instantaneous merge operations in Figure 5.8, and for stop and pause actions[9]. This is equivalent to assuming that in SAM, each user is dynamically allocated a portion of the synch buffer of size up to the CPE-buffer size in our scheme. Our results indicate that the fractions of blocked and discontinuous actions are dramatically increased in the SAM case (up to 33% higher for blocked actions), confirming

[9] Note that our experiment slightly improves the SAM protocol in comparison with the scheme originally proposed in [64], by considering scalable batching, and by isolating respective allocations of B- and I-channels. The blocking probability documented in [64] is linear in traffic intensity.

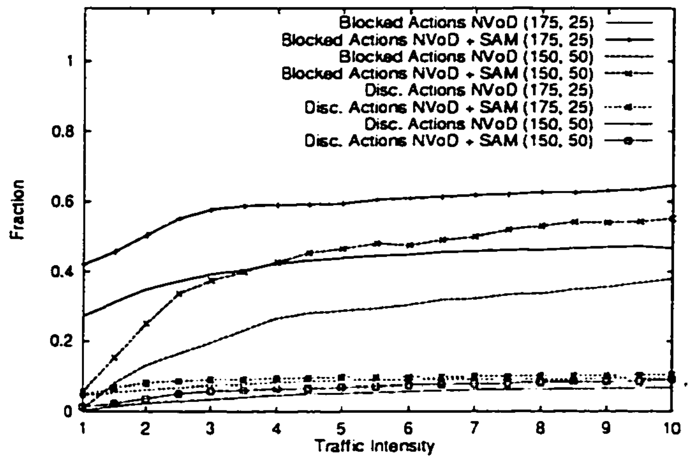


Figure 5.18: Average fraction of blocked and discontinuous VCR actions: comparison with the SAM protocol.

that active use of the CPE buffer to serve CPE-actions and to shorten merge operations, and support for discontinuous actions, are two very effective ways to improve customers' QoS. These facts were confirmed experimentally: lengthened merge operations tend to reduce the availability of I-channels, which, in turn, decreases the number of I-actions thus increasing the average I-channel holding time. Lastly, dispersions of blocked VCR actions and I-actions also showed that scarcity of I-channels leads to greater unfairness when the SAM protocol is used, by favoring customers viewing the most popular movie.

5.4.3 Partially-Patient Customers

Providing service to partially-patient customers imposes constraints on the choice of a batching policy. In order to reduce cost, it is more important for a multicast VoD server to achieve a high throughput by servicing a large number of (non-defecting) customers, than to provide a low latency service at the cost of high defection rates. By admitting customers at a higher rate, however, an efficient VoD server may degrade the QoS received by the same customers during VCR actions since a heavier traffic load would be admitted into the system. In this section, we study the tradeoff between efficient batching and degradation of QoS in VCR actions. We also consider the dual problem of sensibly partitioning the channel capacity of the VoD server between B- and I- channels.

5.4.3.1 Choice of a Batching Policy

As in Section 5.4.2, we assume that a VoD server has to choose a batching policy among NVoD, QVoD-enhanced NVoD, and QVoD. In NVoD, partitioning the B-channel capacity to minimize defection rate results in extremely unfair configurations, in which most channels are allocated to two or three most popular movies. We therefore considered simple heuristics such as the allocation of channels in proportion to movie popularities which achieve a good tradeoff among defection rate, average latency, and fairness in both variables in NVoD and correspondingly, in QVoD-enhanced NVoD (with the set of thresholds minimizing customers' defection rate). As for QVoD, the threshold should be chosen such that the defection rate is minimized. Consequently, we compared the following six systems:

1. An NVoD server with $[s_1, \cdots, s_N]$ minimizing the average NVoD phase offset; this configuration is called *NVoD EW-OPT*. This configuration yields minimum latency in the case of infinite patience, by allocating B-channels more evenly among movie titles. It is used, for comparison purposes, to determine whether it is also effective with partially-patient customers.

2. A QVoD-enhanced NVoD server with the same channel allocation vector $[s_1, \cdots, s_N]$ defined by NVoD EW-OPT, used in conjunction with $\overline{K}^{opt}$ which minimizes the defection rate for each movie title. This configuration is called *QVoD-enhanced NVoD EW-OPT*.

3. An NVoD server with the partition $[s_1, \cdots, s_N]$ making an acceptable tradeoff among throughput, phase offset and fairness. We showed in Chapter 3 that allocating channels proportionally to the popularities (*NVoD T-PROP*) is preferable for very impatient customers.

4. A QVoD-enhanced NVoD server with the same partition $[s_1, \cdots, s_N]$ determined by NVoD T-PROP. This configuration is called *QVoD-enhanced NVoD T-PROP*.

5. An NVoD server with $[s_1, \cdots, s_N]$ maximizing the NVoD throughput; this configuration called *NVoD T-OPT*, is used for comparison purposes, as it indicates the maximum throughput achievable by an NVoD server. NVoD T-OPT tends to allocate all channels to the most popular movies as the patience factor decreases.

6. A QVoD server whose threshold K^{opt} is chosen so that customer's defections are minimized. We also use the LCFS scheduling, better suited to very impatient customers.

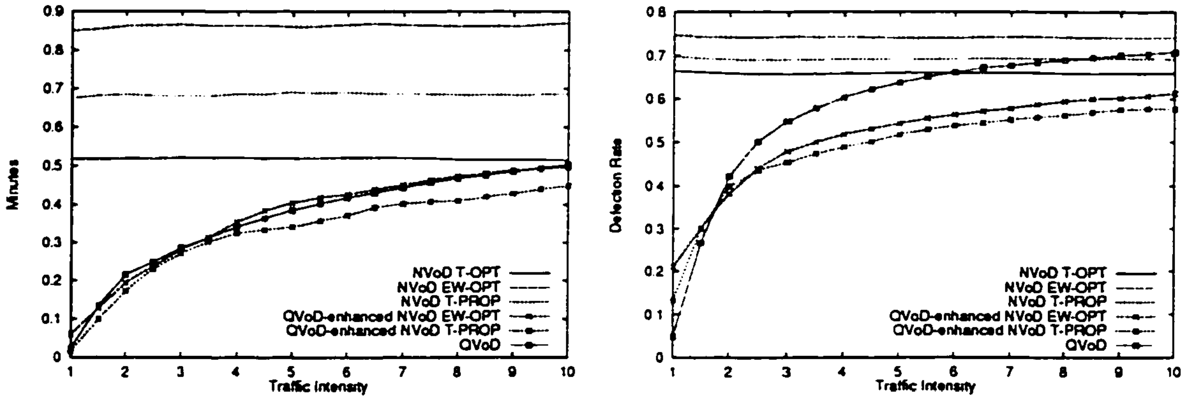


Figure 5.19: Average admission latency and defection rate for $\beta = 0.01$.

Figure 5.19 represents average admission latency and defection rate for 200 B-channels and a patience factor $\beta = 0.01$, corresponding to customers who are willing to wait an average of 1 minute, thus very impatient. Note that this patience behavior is chosen for illustration purposes in an extreme case of customers' behavior. In practice, customers may be more patient, and correspondingly, the defection rates presented in the second graph of Figure 5.19 considerably reduced, along with the discrepancy among these defection rates. It is clear that QVoD-enhanced NVoD batching policies outperform NVoD systems, by lowering defection rate and average latency by at least 25%. Surprisingly, NVoD EW-OPT shows the highest latency. Note that the average latency can be a somewhat misleading metric since it is calculated based on requests that actually get serviced. QVoD performance, while not as good as QVoD-enhanced NVoD, is still a better alternative than NVoD, except in an unpractical situation where NVoD T-OPT is used for high loads. We also found that the dispersion of the defection rate in QVoD-enhanced T-PROP was reduced by half as compared to NVoD T-OPT, and the dispersion of the average latency was comparable to that of NVoD EW-OPT. Our point is clearly illustrated: QVoD-enhanced T-PROP appears to be the most adapted policy to customers' admission.

We consider next, for $(B, I) = (200, 0)$, the breakdown among VCR actions (Figures 5.20 and 5.21). When no I-channel is provided to support VCR actions, the difference in throughput among batching policies does not affect the QoS received by customers during CPE-actions and discontinuous VCR actions, which does not depend on the system load. Thus, the differences observed in Figures 5.20 and 5.21 are due to the schedule generated by each batching policy and can be explained in the same way as we explained the slightly better performance of QVoD with respect to discontinuous operations in the case of infinitely-

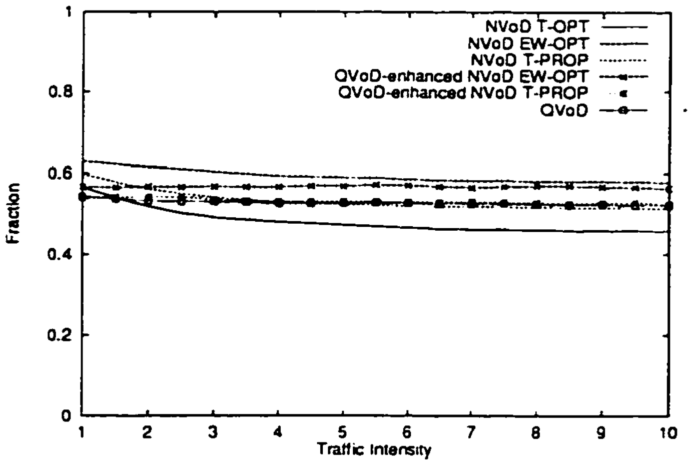


Figure 5.20: Average fraction of blocked VCR actions for $(B, I) = (200, 0)$ and $\beta = 0.01$.

patient customers (cf. Figure 5.12). Discontinuous operations are mostly backward actions (pause, stop, and rewind), which extend the average connection time experienced by customers. NVoD T-OPT allocates most channels to most frequently-requested movies, thus reducing the discrepancy observed in discontinuous actions. A lower discrepancy in NVoD T-OPT tends to reduce the extension in program subjective duration caused by backward-dominated discontinuous actions. Since the absolute values of the average duration in pause and stop actions, and of the average jump sizes in fast forward and rewind actions are unchanged, the relative fraction of discontinuous actions will then be greater in NVoD T-OPT, as illustrated in Figure 5.21. The fraction of CPE-actions — which mainly depends on the relationship between CPE-buffer size and interaction durations — remains unchanged, and the fraction of blocked VCR actions will finally be reduced accordingly. The opposite is true for NVoD EW-OPT, for which the B-channel allocation, albeit more evenly distributed, leads to higher discrepancy in discontinuous actions. In between these two extreme cases, QVoD-enhanced NVoD T-PROP only yields a 10% increase in blocked actions in comparison with NVoD T-PROP.

When a portion of the channel capacity is dedicated to I-channels, efficient batching will increase the load on I-channels, thereby reducing customers' QoS in VCR actions. The tradeoff between high throughput and I-channel availability is a clearcut throughout Figure 5.22, which represents the fraction of I-actions in two partitions of the channel capacity, $(B, I) = (175, 25)$ and $(B, I) = (150, 50)$). Since discontinuous actions were less than 10% of all VCR actions, and the fractions of CPE-actions were identical in all cases, Figure 5.22 also indicates the relative performance of the various batching policies in terms

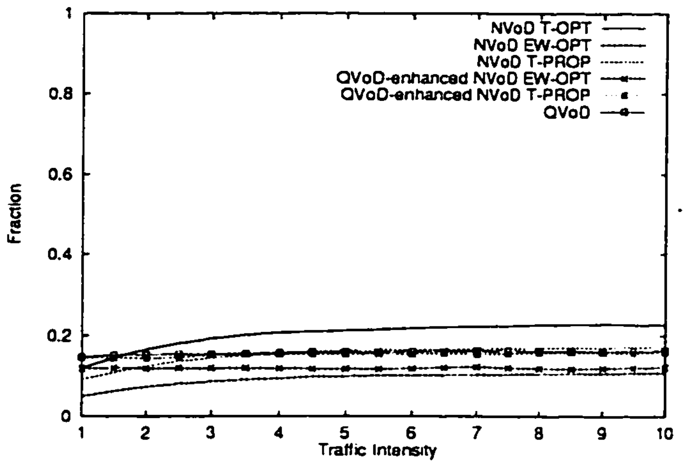


Figure 5.21: Average fraction of discontinuous actions for $(B, I) = (200, 0)$ and $\beta = 0.01$.

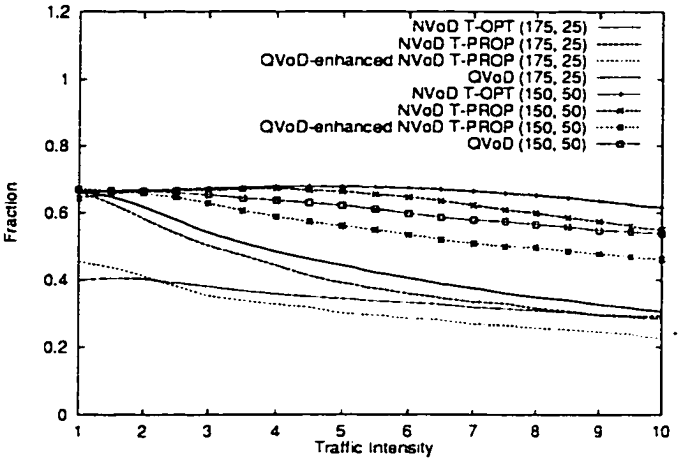


Figure 5.22: Average fraction of I-actions for $\beta = 0.01$.

of blocked actions. Although a choice has to be made between admission and interactive QoS, the increase in blocked actions incurred by QVoD-enhanced NVoD T-PROP is only of about 10% in comparison with other realistic policies such as NVoD T-PROP and QVoD. In comparison with a 25% improvement in admission QoS, among all systems studied, QVoD-enhanced NVoD T-PROP offers the best overall performance. Similarly to the infinitely-patient customers case, the choice of a batching policy is mainly determined by admission performance.

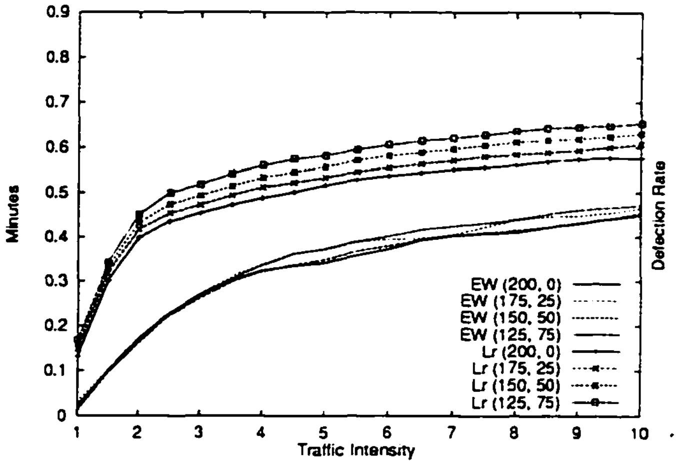


Figure 5.23: Average admission latency and defection rate for QVoD-enhanced NVoD T-PROP ($\beta = 0.01$).

5.4.3.2 Choice of a Partition between B- and I- Channels

If more B-channels are allocated, a greater throughput and, possibly, lower admission latency, can be achieved at the cost of a higher VCR actions blocking probability. We investigated this tradeoff in four different partitions, $(B, I) = (200, 0), (175, 25), (150, 50)$ and $(125, 75)$. Figures 5.23 and 5.24 represent both admission and service performance in QVoD-enhanced NVoD T-PROP. As can be seen from Figures 5.23 and from the first graph of Figure 5.24, partitions (150, 50) and (125, 75) dramatically reduce the fraction of blocked and discontinuous actions (for instance, from 55% of blocked actions in (200, 0) to 0% in (125, 75)), with (1) minor impact on the defection rate, (2) no change in the average latency, and (3) no loss in scalability.

Consequently, although there exists a tradeoff, the choice of a partition is considerably more critical to QoS in interactive service than to customers' admission. More importantly, the quasi-parallelism that can be observed in each figure for the same batching policy, among curves corresponding to different partitions, indicates almost no loss of scalability caused by changes in the breakdown between B- and I- channels. Clearly, a multicast VoD server can provide unrestricted support for interactive operations with minor impact on system scalability *and* performance. Although it is not possible to detail here each particular case of channel capacity, movie selection, and customers' patience, all our simulations confirmed this conclusion.

In addition to sensible partitioning, a VoD server may also protect customers from QoS degradation by selectively discarding customers' requests to reduce the load on I-

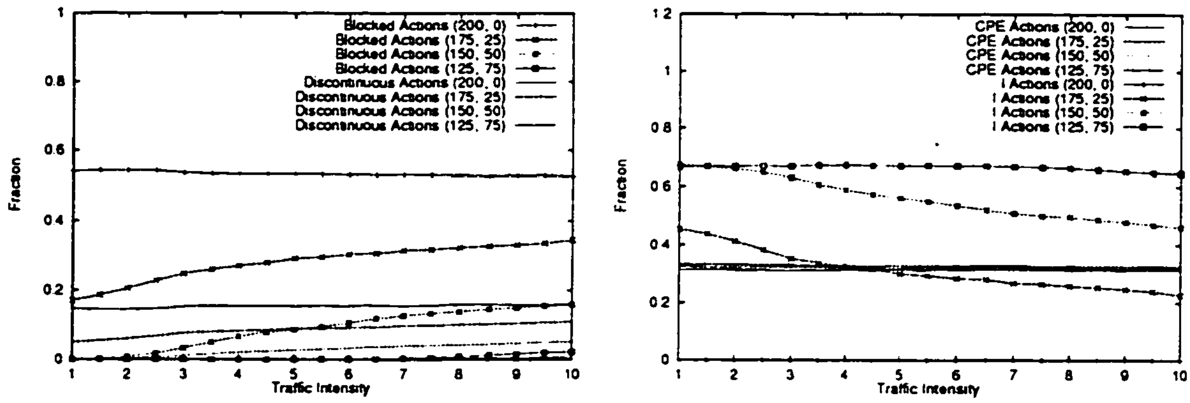


Figure 5.24: Average fraction of blocked, discontinuous, CPE- and I- actions for QVoD-enhanced NVoD T-PROP ($\beta = 0.01$).

channels. This approach, although attractive when variations of multicast group sizes are caused by time-varying arrival rates or non-static behavior, is not well-adapted to scalable systems. Figures 5.23 and 5.24 illustrate this point in various cases of partitions (e.g., $(B, I) = (175, 25)$), since discarding enough customers' request to reduce the traffic intensity from 10.0 to 5.0 only decreases the fraction of blocked actions by less than 10%.

5.4.4 Influence of System Parameters on VCR Functionality

For completeness, we summarize in this section other important simulation results on the VCR functionality for various CPE-buffer sizes, channel capacities, and interactive behaviors. The corresponding empirical studies corresponding to this summary are detailed in Appendix F.

CPE-Buffer Size: Changes in the fraction of I-channels and in the CPE-buffer size are found to affect the partition of VCR actions in an independent fashion. More precisely, increasing the CPE-buffer size for an arbitrary partition (B, I) only marginally affects the number of I-actions, even though we found that the probability of merge success increases almost in proportion to the increase in buffer size. Correspondingly, increasing the fraction of I-channels for a fixed CPE-buffer size will not affect the fraction of CPE-actions. Also, for high traffic intensities, increasing the CPE-buffer size (as opposed to increasing the fraction of I-channels) is usually a more effective way to maintain scalability.

Channel Capacity: If the fraction of I-channels remains constant, the variations of customers' QoS are shown to be insensitive to changes in the channel capacity. This property is another illustration of our scheme's scalability. The support for discontinuous operations is also found to be more effective with frequently-requested movies. for which more channels are allocated. This observation can be used to provide a more equitable service by allocating more I-channels to less popular or "cold" movies, for which the average phase offset is greater than the CPE-buffer size. Albeit counter-intuitive. this approach can effectively prevent customers viewing the most popular movies from saturating the I-channel capacity by unnecessarily using I-channels for pause. stop. fast forward and rewind actions; these same VCR actions could have been served discontinuously with very low, and therefore acceptable, discrepancy.

Interactive Behavior: From our numerical study with various CPE-buffer sizes, changes in the duration of VCR actions are found to affect only the fraction of CPE-actions, while the fraction of I-actions remains constant. For longer actions. the loss in CPE-actions is mostly translated into a higher fraction of discontinuous actions of reduced discrepancy. whereas the fraction of blocked actions remains relatively unaffected. Also, a higher level of interactivity is likely to generate more demand on the CPE buffer and on the pool of I-channels. We actually observed that the fraction of CPE-actions only slightly decreases when customers' level of interactivity increases. thus showing that the CPE buffer is a robust support for continuous service of VCR actions when customers' level of interactivity varies. In any case, support for discontinuous operations is usually found effective in graceful QoS degradation when duration or level of interactivity of VCR actions are not precisely known. or increase randomly.

5.4.5 Discussion

In summary, simulations show that the proposed scheme for supporting VCR functions. when used in conjunction with scalable batching policies. provides (1) scalability in admission QoS, (2) scalability in interaction QoS. (3) graceful QoS degradation in VCR actions when interaction or admission behaviors vary. (4) robust continuous service of VCR actions. In addition, the breakdown among VCR actions is generally independent of the batching policy, provided that the latter is chosen among NVoD-based systems. This independence between admission and interaction QoS is very advantageous as it allows to protect QoS in VCR actions from workload fluctuations and overloading situations.

Having, so far, focused on monolithic storage organizations. an important issue is to determine the efficiency of our proposed framework for VCR actions when used in clustered disk arrays, and more particularly in conjunction with ad-hoc batching policies such as NVoD-h. This issue is critical for three reasons: (1) NVoD is not always feasible in disk arrays, (2) QVoD-enhanced NVoD requires extensive tuning, which depends on patience, time variations in arrival rates, and popularity models; and (3) optimal NVoD or QVoD-enhanced NVoD schedules are not always feasible in clustered disk arrays, in which the maximum number of channels possibly allocated to a movie title depends on the number of clusters holding this particular movie title.

As we have seen in this section, QoS in VCR actions remains largely unaffected by whether B-channels are sourced at regularly spaced intervals or not: it depends mostly on the number of channels allocated to each movie title. In clustered disk arrays, the maximum number of channels that can be allocated to each movie title depends on number of clusters, movie selection, and channel capacity. However, if room is available for proportional replication (e.g., as in constraints of type **C**, or, possibly, of type **A**), results obtained in Chapters 3 and 4 with moderately-patient customers indicate that NVoD-h performs comparably to NVoD and QVoD-enhanced NVoD in terms of customers' admission QoS, and therefore in terms of B-channel allocation. In this case, since customers' QoS during VCR actions is relatively insensitive to the choice of a batching policy, we can state that full-fledged, yet scalable support for VCR actions is feasible with NVoD-h in clustered disk arrays. Future work will therefore determine which constraints on channel capacity, movie selection, number of clusters, and CPE-buffer size are more favorable to a fully-interactive multicast VoD service. Whatever the case may be, we have provided a framework to investigate these issues in a systematic manner.

Another important issue is to determine an adequate partition of the channel capacity between B- and I- channels, possibly dynamically as the VoD server workload changes, for instance due to nonstationary request arrival rates, movie popularities, or customers' patience. One approach, initially proposed in [31], proceeds incrementally by first allocating a small number of I-channels and increasing the size of the I-channel pool until an acceptable tradeoff between customers' QoS and server throughput is found. Partitioning can also be studied in terms of economics, since admission and VCR actions may be charged differently, and consequently, B- and I- channel usage may generate different revenues. This issue is further complicated if customers are charged differently for a same VCR action depending on its location within the program. For instance, customers skipping advertisements could be charged more since these customers actually reduce the effectiveness of the VoD service subsidies.

A dynamic partitioning policy should ideally control on-line the size of the I-channel pool to cope with the changes in the workload not only in terms of video access pattern, but also in terms of the surges in VCR-control activity and request arrival rates. In addition, for a fixed number of B-channels (or, alternatively, a fixed fraction of the bandwidth allocated to broadcast service), the number of available I-channels varies dynamically, depending on the delivery rate required by the VCR actions served by the I-channels in use. Partitioning is therefore a very complex problem as it is usually impossible to express key performance variables such as customers' QoS and server throughput in a closed-form that could be used in partitioning heuristics. This observation holds even in the simplest cases of interactive functionality, as shown in [31] when customers only perform pause actions. Determining accurate, dynamic partitioning heuristics is therefore left as future work. It should be noted, however, that this issue is less critical within our framework for VCR functionality since (1) scalability in both admission and interactions ensures resilience to nonstationary request arrival rates, and (2) as we saw in Section 5.4.3, the partition between B- and I- channels need not be specified accurately in order to provide acceptable QoS.

5.5 Active CPE-Buffer Management

The CPE-buffer management presented thus far can be considered *passive* because no direct attempt was made to change the relative position of the play point within the CPE buffer. By default, the only mechanism available to the CPE buffer is to discard past frames at a rate slower or faster than the arrival rate of frames from the server, while receiving frames from the current channel in a synchronous fashion. Thus, the position of the play point relative to the most recent frame remains unchanged unless an explicit action is taken. Even when changes are explicitely required during discontinuous or merge operations, multicast group changes will be made so as to leave as many future frames as possible in the CPE buffer thus avoiding bandwidth waste in duplicate transmissions. Moreover, the duration of merge operations is decreased if the CPE-buffer management keeps as many future frames as possible while the customer is served by an I-channel. In a sense, passive management implicitly favors forward VCR actions by keeping future frames whenever possible. We would like to propose and evaluate mechanisms for better CPE-buffer management when interaction behavior is dominated by backward interactions, or when backward interactions are as likely as forward interactions.

5.5.1 Preventive Multicast Group Change

Customers' interaction latency, blocking probability and load on the pool of I-channels can be reduced if a large fraction of VCR actions are satisfied by the CPE buffer, without support from the VoD server. The underlying idea behind active CPE-buffer management is to dynamically adjust the relative position of the play point within the CPE buffer so the probability of successful CPE-actions may be enhanced. Efficient CPE-buffer management is highly dependent on customers' behavior. For instance, if backward interactions are as likely as forward interactions, it would be sensible to keep the play point as close to the middle of the CPE buffer as possible, so that past frames and unplayed frames are equally available for access. If interactive accesses are dominated by backward interactions, the playout point should rather correspond to the most recent frame.

Let's first focus on an interaction behavior dominated by backward interactions. As illustrated in Figures 5.3 and 5.7, it is possible to take advantage of discontinuous pause or stop action and of merge attempts to make additional buffer space available for past frames. This is done by switching the customer into the multicast group whose play point is ahead and close to the customers' play point. Furthermore, if the customer stays long enough in play mode while being served by a B-channel, a *threshold-activated preventive merge operation* (TAPMO) can be initiated. The basic idea behind a TAPMO is to disregard unplayed future frames in order to make buffer space available for past frames. In a TAPMO, an attempt is made to switch the customer to a B-channel whose play point is closer to the customer's play point than that of the current B-channel, and ahead, only after a threshold number of future frames are accumulated in the CPE buffer. This is illustrated in Figure 5.25, in the case of threshold set to one fourth of the CPE-buffer size. A TAPMO is basically a merge operation similar to that represented Figure 5.7, and hence, does not require any additional support other than merge functionality. It is performed instantaneously, by (1) locating a target channel among candidate channels (steps 1 and 2 in Figure 5.25), and (2) disregarding as many future frames as needed to be in synchronization with the target B-channel (step 3).

If backward interactions are as likely as forward interactions, candidate channels can be chosen so the play point is kept as close as possible to the middle of the CPE buffer. This is illustrated in Figure 5.26, where the target channel is the candidate channel whose play point is the closest to the *target point*, located half a buffer size ahead of the customer's play point.

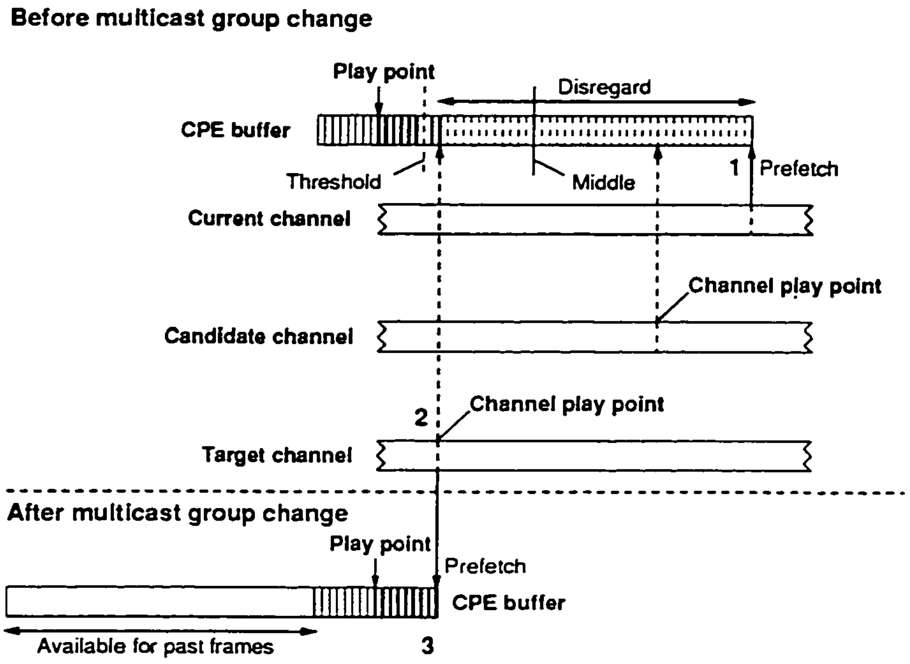


Figure 5.25: Forward adjustment of the CPE-buffer play point.

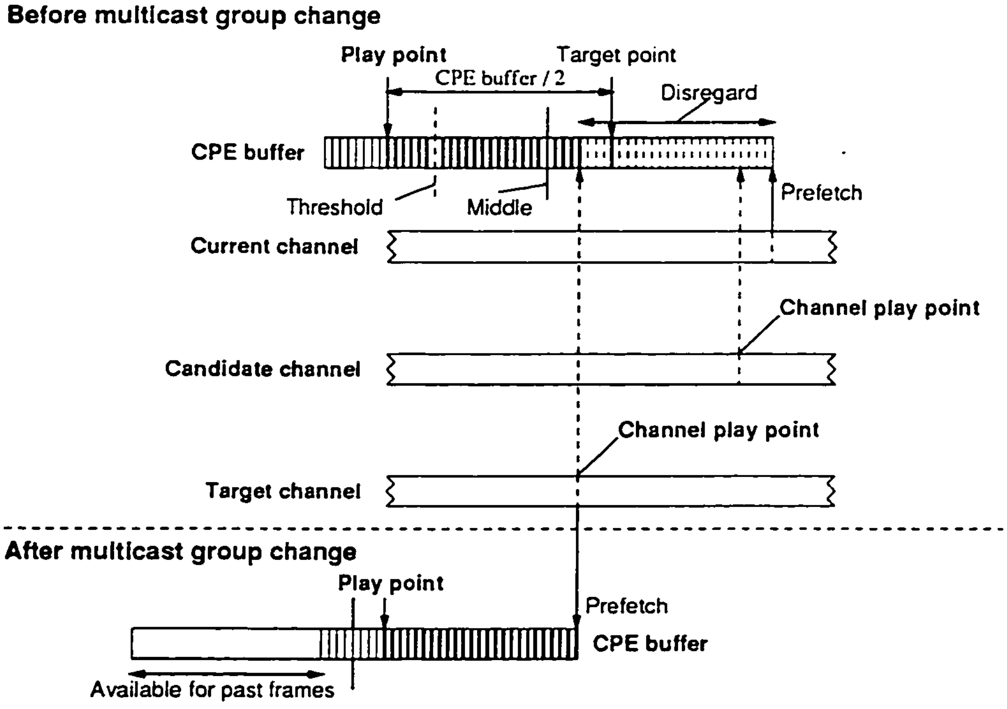


Figure 5.26: Middle adjustment of the CPE-buffer play point.

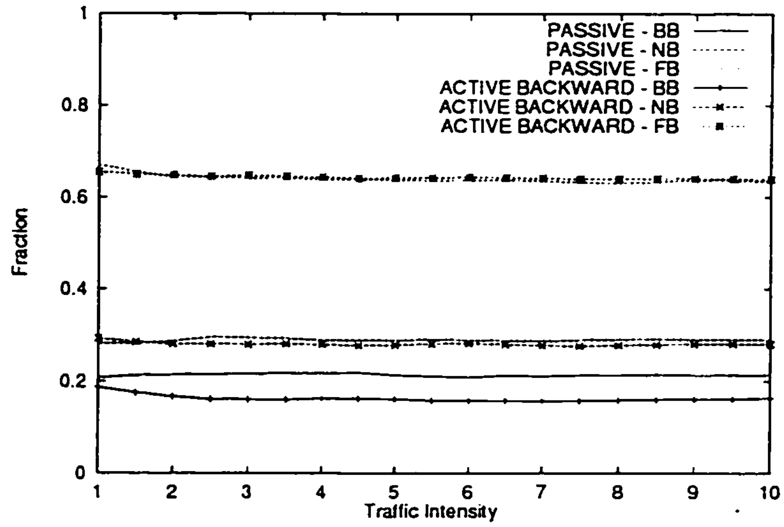


Figure 5.27: Active CPE-buffer management: average fraction of blocked VCR actions.

5.5.2 Numerical Results

The effectiveness of our proposed scheme was evaluated by comparing passive and active management for the three biased behaviors introduced in Table 5.1, namely, backward, neutral, and forward bias. For an illustrative purpose, we want to show that "active backward" management as described in Figure 5.25 can improve customers' QoS when VCR actions are biased towards backward actions, and leave customers' QoS relatively unaffected in other cases. Active buffer management is effective whenever a TAPMO is likely to succeed, that is, when the CPE-buffer size is relatively large in comparison with both the duration of VCR actions and average phase offset between consecutive start times of the same movie. The latter condition will certainly hold in a large-scale VoD system (e.g., 10,000 channels serving 100 titles). For tractability, we assumed a system of capacity 200 B-channels dedicated to 5 movie titles, a CPE-buffer size of 5 minutes' worth of video data, and VCR actions of average duration of 1 minute. The group change threshold was set to 0.5.

As can be seen from Figure 5.27, active buffer management reduces the number of blocked actions by 25% by serving more actions with the CPE buffer. A relative reduction of 50% in discontinuous actions is also found, although these actions only represent less than 5% of all VCR actions when interaction behavior is biased towards backward actions. The slight improvement noticed for a "neutral" bias corresponds to the fact that neutral bias is itself slightly biased towards backward actions, due to pause and stop actions. This does not invalidate our conclusions. Lastly, similar results were obtained in separate simulations for a group change threshold set to 0.25. The exact specification of an optimal threshold is therefore not critical, provided it is set to a reasonable value (e.g., ≤ 0.5).

5.6 Conclusions

In this chapter, we (1) identified efficient mechanisms to provide full-fledged, yet scalable support for VCR actions in multicast VoD systems, and (2) evaluated them by considering both the VoD server performance and customers' QoS under various realistic scenarios. First, using the CPE buffer, multicast VoD servers are shown to be able to provide discontinuous and intermittently-continuous VCR actions. Next, we showed that I-channels can be used synergistically with the CPE buffer to provide unrestricted interactive operations. Our analysis extends the work in [12, 31, 64] by improving scalability and QoS in both admitting a new request and servicing interactive operations. Our numerical results showed that QVoD-enhanced batching policies consistently outperform other scalable policies in admitting customers, without any significant loss of QoS in VCR functionality. We also found that as customers' patience decreases, a larger fraction of I-channels can be dedicated to interactive service with a minor impact on system scalability and admission QoS. Comparisons with the SAM protocol confirmed that both use of the CPE buffer and support for discontinuous actions are very effective in improving customers' QoS. Hence we studied the advantages of our framework for VCR functionality for various CPE-buffer sizes and channel capacities. As an alternative to I-channels, the CPE mechanisms for continuous and discontinuous operations provide effective support for adapting QoS to changes in CPE-buffer size, channel capacity, and customers' behavior in both admission and interactions (frequency or duration of requests for VCR actions). Finally, we presented the concept of active CPE-buffer management, which is shown to improve customers' QoS when the interaction behavior is biased towards a particular type of VCR action.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This dissertation has established a framework for specifying long-term resource allocation in VoD end-systems so that a large collection of movie titles and channels can be accessed by a maximum number of users with service scalability in interactions, and at the lowest cost. This issue is becoming increasingly important, given that economical viability and future large-scale deployment of digital video services depend on customers' appreciation of the service delivered. We now recapitulate the primary contributions of this dissertation, and suggest future work.

6.1 Research Contributions

We identified and solved scalability bottlenecks in realistic, challenging settings, in order to determine how to provide an increasingly large number of customers with an increasingly interactive service. This research has made contributions in the following areas.

Storage organizations for VoD service: We showed that hybrid storage organizations, in which videos are partitioned into several disk-array-based clusters and striped across each disk array using CGS, should be chosen for cost-effectiveness, low admission latency, and high availability. We introduced the concept of *eligible clustered configurations*, and proposed an algorithm of low, polynomially-bounded complexity for optimal video allocation across clusters. This algorithm provides a useful alternative to the heuristics that have been commonly proposed to solve such allocation problems. It can also be applied to many practical situations where objects are statically assigned to storage and accessed with heterogeneous frequencies. To provide further guidance in the choice of a storage organization, it was important to determine at what cost and additional support D-VoD could be upgraded to T-VoD. We addressed this issue by proposing a probabilistic connection admission control based on a model of channel usage in T-VoD. Our simulation results indicated that, for a reasonable range of customers' behavior. D-VoD can be upgraded to T-VoD at low cost. It was also important to identify practical situations in which the OMPD scheme could be chosen, as this scheme is easier to deploy, upgrade, and experiment with when the service is expanding. We showed that even though it is often regarded as a naive approach to storage configuration, OMPD with adequate OCAP specification may actually be acceptable for highly reliable service of a small number of movies to a large user population, especially when customers' behavior is only approximately known. This way, movies can be rearranged and replicated as needed during off-peak hours. and thus made available for viewing during peak hours based on anticipated demands.

Optimal scheduling in NVoD systems: We presented an analytical, yet realistic approach to program scheduling in NVoD systems. which are becoming increasingly popular among computer manufacturers and cable companies. We integrated customers' and service provider's points of view in a mathematical framework and derived the optimal schedule of movies of different popularities for maximum throughput and the lowest average phase offset. Such derivation is independent of the time variations in the request arrival rates: it only depends on the number of channels available. customers' patience. and on the performance variable which is most valued by the NVoD service provider. i.e.. throughput. average phase offset, fairness, or a tradeoff among all of these. We also analyzed QVoD. which schedules channels based on a threshold of requests. By using QVoD in conjunction with NVoD in the so-called QVoD-enhanced NVoD. we improved the throughput without compromising customers' QoS. in terms of average phase offset and the corresponding dispersion. This result is important since the problem of retrieving videos *exactly periodically* from disk arrays in a cost-effective way requires time-consuming. often only near-optimal heuristics. While unlikely in a wide range of cases. if it has been determined that an NVoD schedule is not feasible, a VoD server can either (1) change the retrieving cycle length. until the schedule is found feasible; (2) adopt another less efficient and cost-effective storage configuration or organization such as OMPD; (3) delay channels until an available slot opens up: and (4) use another scalable batching policy. The first two cases are costly and not practical, since they require adding resources and changing the layout of videos on disks. In the third and fourth cases, channels can be sourced quasi-regularly as in QVoD-enhanced NVoD yet provide a throughput higher than in NVoD. as long as each movie title is allocated the pre-determined number of channels. This is particularly useful when the popularity profile changes throughout the day, in which case an NVoD schedule could become invalid unpredictably. A natural question that arises is whether it is possible to design heuristic algorithms that take advantage of scheduling flexibilities in (3) and (4) for cost-effectiveness. In particular, future work will determine to what extent QVoD-enhanced NVoD simplifies retrieval scheduling in disk arrays. Moreover, we also found that simple heuristics such as the allocation of channels in proportion to movie popularities yield good results, thus allowing for more freedom in the choice of a schedule. Our theoretical study thus provides (1) bounds on the system performance, and (2) ad-hoc solutions to approximate these bounds, which are directly applicable to real systems.

Scalable batching policies: After focusing on the commercially available NVoD, it was important to systematically identify and compare scalable batching schemes in disk-array-based VoD servers, by considering factors such as customers' behavior, storage organization, and time variations in the request arrival pattern. When batching is done without any control in the channel allocation process, three factors are found to make the system prone to channel clumping, which can then cause short- and long-term congestions. These factors are (1) the inherent lack of variability in movie lengths, (2) the access locality to movie titles, and (3) the small number of different movie titles relative to the number of channels allocated to them. We illustrated this sensitivity to congestion cycles with the "on-demand" batching policy, well-known for its lack of scalability, and under specific "worst-case" workloads. We next introduced a new metric called the burst absorption capability (BAC). We used this metric to comparatively evaluate various scalable batching policies. The simplest scalable batching policy, NVoD-h, is found to vastly improve on-demand channel allocation, while being easily implementable in both monolithic and clustered disk arrays. This result is quite useful as we recognized the limitations of NVoD-based batching in both monolithic and clustered disk arrays, and the need for ad-hoc scalable batching alternatives. Even though support for discontinuous VCR actions in NVoD-h is only provided in an average sense, different schemes (summarized next) can be used to support unrestricted VCR actions without compromising the system scalability.

Scalable fully-interactive VoD service: Although most underlying ideas are not new, our work is the first to provide a detailed, thorough engineering presentation and a quantitative evaluation of mechanisms to provide full-fledged, yet scalable support for VCR actions in multicast VoD systems. From a customer's point of view, there is a clear advantage in having more functionalities in the service provided. From a service provider's point of view, VCR actions can generate an extra revenue if customers are charged based on the additional resources requested. More specifically, we showed that, unlike shared buffers located closer to the server, small individual CPE buffers can be used in synergy with dedicated channels to allocate and reclaim resources more efficiently than existing schemes, and therefore provide unrestricted VCR functionality. For evaluation purposes, we used an idealized, yet realistic model of customers' admission and interaction behavior to capture several key interaction patterns. Our experimental results, restricted to monolithic disk arrays but applicable to clustered storage configurations, showed that QVoD-enhanced batching policies consistently outperform other scalable policies. We also found that the CPE mechanisms for continuous and discontinuous operations provide effective support for various CPE-buffer sizes, channel capacities, and VCR actions characteristics. We finally introduced the concept of *active CPE-buffer management*, which is shown to provide effective support for QoS and graceful QoS degradation in case of resource shortage, or when interaction behavior is biased towards a particular type of VCR action. As abandoning closed, proprietary STB standards in favor of open standards becomes a reality (see [14] for a complete discussion of this issue), our work can provide guidelines as to what minimum hardware cost (memory and processing power) is needed for scalable interactive service. Also, since integrated and simple chipset solutions exist for the hardware of first-generation set-top devices, the emphasis of STB development is likely to shift to the architecture of the software required for these products. Active buffer management and CPE/server synchronous communication can potentially enhance the functionality of STBs as they become integrated into more powerful "home appliances" or personal computers. (A recent discussion on the software design issues in STBs can be found in [27, 40].) Lastly, as mentioned in Section 5.3.1 of Chapter 5, future work will determine whether a shared-buffer approach is preferable to decentralized management of non-shared buffers. (It should be noted that both schemes are not mutually exclusive: a VoD server could allocate a synch buffer for bufferless clients, while customers with a CPE buffer would use our scheme.) Until then, a more powerful, albeit more expensive STB, will alleviate the load on the VoD server, which can, in turn, provide better service scalability and therefore, a cost-effective, expandable service.

6.2 Future Directions

In addition to the various open issues that were discussed throughout this dissertation, we now present some research problems for direct continuation of our work.

We made several assumptions to illustrate general guidelines for practical and simplified scenarios. First, various aspects of consumer use of continuous digital media, such as patience, time variations in the request arrival pattern, movie popularities, and models for VCR actions, are inherently speculative and mostly justified for their intuitive realism. These behavioral models can be further refined, for instance by changing the set of popular movies over the course of a day, or by considering other nonstationary and patience models. Nevertheless, until field data confirms their pertinence, our assumptions and their proper are inevitable to (1) glean heuristics that will work well in practice, and (2) identify or anticipate the human factors and worst-case scenarios that will challenge system scalability. This dissertation is a first step towards achieving both goals. We suggested in Chapter 5 an experimental setup for monitoring VCR actions in visualization labs. This example illustrates that throughout VoD deployment, the same emphasis should be put on both modeling and monitoring human/media interactions. This twofold effort requires appropriate tools located in end-systems and delivery networks in order to support iterative refinements through implementation and experimentation.

Most of the results presented in Chapter 2 are applicable to a wide range of commercial disks with different capacity and bandwidth. A next logical step is therefore to generalize our study to different disk parameters, movie lengths, video rates, and applications imposing different constraints on the storage organization. For instance, a video server could archive videoconferences on disks, and later serve these stored objects on an on-demand basis. Stored videoconferences differ from movies in several ways. First, they may not be accessed according to the same popularity profile as movies. Second, frame rate, screen size, encoding standard, and session duration may all change from one videoconference to another. Third, although content insertion may not be used in such applications, storing videoconferences actually requires a high versatility since data may be written as frequently as it is read. Lastly, viewers may be more likely to scan the data in order to access specific segments of the videoconference. These different requirements make it difficult to determine an adequate "multi-purpose" storage organization.

Another issue more relevant to a movie-on-demand environment is that it may be more practical to reconfigure a local disk-array-based VoD server from a local movie archive, instead of accessing a metropolitan archive server. This archive would be used to download, in an asynchronous fashion, new material from larger metropolitan archives for later reconfiguration, or to store less popular movies locally. Dimensioning and managing a local storage hierarchy based on secondary and possibly tertiary storages (cf. Section 2.1 of Chapter 2) are open issues. Also, by partitioning disk arrays into equal-size clusters, we were able to infer an optimal video allocation, which was then used to evaluate support for service availability. It is clear that the more general case of uneven clusters is more flexible. but the VoD server must ensure that clusters are accessed in proportion with their relative channel capacity. In practice, the video allocation algorithm presented in Chapter 2 can be used when cluster sizes are known. Otherwise, more sophisticated heuristics should be investigated.

Other assumptions were made for illustrative purposes in Chapter 5 on video transport and support for interactions. For a simple description of CPE-buffer management, all frames were assumed to be of the same size and VCR actions did not incur loss of frames. However, most of the mechanisms presented in Sections 5.2 and 5.3 can be easily adapted to VBR video or to CBR transport of video with different frame sizes and interaction mechanisms. Moreover, the comparative evaluation results presented in Sections 5.4.2 and 5.4.3 should remain unchanged, provided all batching policies use the same video encoding and transport, for playback or interactions. Future work will determine whether CPE buffer and I-channels can be used synergistically as efficiently as presented when the assumption of equal-size frames is relaxed. It is worth noticing that partitioning the available bandwidth between B- and I- channels and run-time resource management are more complicated in the case of VBR video, since they depend on the statistical characteristics of the stored video, and on the VCR mechanisms as well. Some guidance may be found in [37] and [67], which investigate the partial support for VCR actions that can be provided when the CPE buffer is used to guarantee starvation-free playback of smoothed (piecewise CBR) *unicast* video. Similarly, more work is needed to determine efficient support for VCR actions of VBR videos in disk arrays and extend the seminal work presented in [88] and [25]. Along with this more realistic setting, additional QoS parameters could be studied, such as the continuity of media streams and the response time to interactive functions (which, ideally, should appear as if the client were operating his/her own VCR).

Finally, although this dissertation addressed scalable resource allocation and usage for large-scale VoD deployment, one needs to answer the basic question: *is there a market for VoD as a service?* Although VoD trials can be traced back to 1985 when Time Warner started the *Orlando project* to test the market, there are, to this day, very few existing trials with more than 2,000 subscribers. Although it is generally agreed that standards must be established that allow competition and growth, by making the initial investment affordable, the low rate of growth in the entertainment industry leads experts to doubt whether anything other than product substitution will occur in the near future. From service providers' and movie studios' standpoints, this is still an attractive option since video stores are usually owned my middlemen who require some cut of the profits, and movie theaters require large investments in real estate. VoD, on the other hand, would just be one more channel of distribution, much cheaper than movie theaters, and which can be integrated into a "price segmentation" strategy by controlling the.timing of release of movies into each distribution channel. From a consumer's point of view, VoD is attractive as a replacement of neighborhood videotape rental stores, given that (1) it has the potential to provide convenient and unrestricted access to a virtually unlimited number of movie titles; and (2) video stores may not have all desired movies and require a customer to make more than one trip to see one movie. It is not clear whether these two advantages are enough to motivate large-scale and costly deployment, unless the service provided to customers is (1) unmatched by current media (e.g., digital HDTV quality with 500 channels or more to one million consumers), (2) made affordable by economies of scale, and additional subscriptions onto cable services. One possible solution to the problem of financing the VoD infrastructure may come from telecommunication and cable companies, which could make the transition by providing new derivative services. For instance, with the capabilities of an STB, an "interactive marketing environment" can be created in which new forms of advertising could send very targeted promotions and allow consumers to actually make a purchase after viewing an advertisement. In a sense, this approach subsidizes the support for more demanding multimedia applications such as VoD. The recent success of on-line shopping over the Internet suggests the viability of this marketing strategy for both advertisers and retailers. The STB can also be used to support pre-scheduled television viewing according to consumers' own schedules and program selections, thus allowing to watch programs which would otherwise be missed. Finally, other possibly lucrative new markets include applications based on bidirectional communication, such as videoconferencing or networked interactive games. Whatever the case may be, service scalability will have to be provided at all levels, from service providers' to customers' equipment.

APPENDICES

APPENDIX A

Enumeration Technique for Calculation of the OCAP

An efficient enumeration technique for the solution of Problem D-VoD OCAP can be obtained by applying the algorithm in [68] (designed for optimal channel allocation for access control of circuit-switched traffic in ISDN environments), which is a version of Fox's algorithm [39, 51]. While a brute-force exhaustive enumeration would require as many as $\binom{N+d.B-1}{d.B-1}$ computational steps, the number of iterations in the algorithm presented below has an upper bound of $d.B$. The basic idea is to iteratively reduce the objective function while restricting each decision variable to at most unit variation, during each iteration. We first define the decrement and increment in the objective function for unit variations in the corresponding server assignment as:

$$D_m = EW(M_{\lambda_m}/D/d_m.B) - EW(M_{\lambda_m}/D/d_m.B+1) \qquad \text{(A.1)}$$

and

$$I_m = EW(M_{\lambda_m}/D/d_m.B-1) - EW(M_{\lambda_m}/D/d_m.B). \qquad \text{(A.2)}$$

The various steps in the integer programming algorithm are summarized in the following heuristic.

Heuristic D-VoD OCAP:

1. Initialization: set $d_m := \lfloor \frac{d}{N} \rfloor, m = 1, \cdots, N-1$; $d_N := d - \sum_{m=1}^{N-1} d_m$. Compute $D_m = EW(M_{\lambda_m}/D/d_m.B) - EW(M_{\lambda_m}/D/d_m.B+1)$ and $I_m = EW(M_{\lambda_m}/D/d_m.B-1) - EW(M_{\lambda_m}/D/d_m.B)$ for $m = 1, \cdots, N$.

2. Begin loop: find $\overline{m}$ such that $D_{\overline{m}} = \max_{m=0}^{N}(D_m)$ and $\underline{m}$ such that $I_{\underline{m}} = \min_{m=0}^{N}(I_m)$.

3. Branch: if $D_{\overline{m}} \leq I_{\underline{m}}$ then go to Step 5 else continue.

4. Update the partition:

- Set $d_{\overline{m}} := d_{\overline{m}} + 1$ and update $D_{\overline{m}}$ according to Eq. (A.1).
- Set $d_{\underline{m}} := d_{\underline{m}} - 1$ and update $I_{\underline{m}}$ according to Eq. (A.2).

Go to Step 2.

5. Termination: Implement $\vec{d}$ as the new OCAP; stop.

It is shown in [68] that the convexity of the objective function enables this algorithm to reach a global minimum in at most $d.B$ steps. This computational efficiency makes it particularly attractive if the popularity of a movie varies over time.

APPENDIX B

Optimal Assignment of Videos to Clusters

In this appendix, we adapt an algorithm initially presented in [57] for dynamic load balancing and optimal retrieval of randomly duplicated VBR video segments in a disk array. Let D be a set of C clusters and M a set of $C \cdot s_C$ instances of videos with partial popularity p'_i to be stored in a clustered disk array. The cluster assigned to video i is denoted by $C(i)$. Clusters' access frequencies are denoted by $F_j = \sum_{i=0}^{C \cdot s_C} p'_i 1_{C(i)=j}, j = 1, \cdots, C$. Our problem is to find an assignment $a : M - D$ such that the highest access frequency among all clusters, $F_{max} = \max_{j \in D} F_j$ is minimized. Let j_{max} denote the index of the corresponding cluster.

Theorem. *Optimum assignment can be found in* $O((C^2 s_C^2 + C^3) \log(C \cdot s_C))$ *time.* □

Proof: Let's consider an arbitrary assignment a. We first list all possible decrements in the access frequency of cluster j_{max}. This list is constructed by replacing each video held by j_{max} by a different video, held by another cluster, and checking whether such a change lowers the access frequency of cluster j_{max}. For each possible decrement d_e, the question whether F_{max} can be lowered by d_e can be reformulated as a *max-flow problem* in a directed network (V, E) constructed as follows. For each cluster j we define a vertex v_j in V, labeled with F_j. For each ordered pair (v_{j_1}, v_{j_2}) in V, we construct the list of all possible decrements in access frequency. We then define a directed edge (v_{j_1}, v_{j_2}) in E, labeled with a capacity $c(v_{j_1}, v_{j_2})$, giving the maximum access frequency assigned to cluster j_1 but can be reassigned to cluster j_2. The capacity of an edge thus gives an upper bound on the flow that may run from one vertex to another. This resulting network is now extended by two additional vertices, a source s and a drain d. For each v_j in V, an edge (s, v_j) is added with capacity $c(s, v_j) = max(d_e + F_j - F_{max}, 0)$, and an edge (v_j, d) is added with capacity $c(v_j, d) = max(F_{max} - d_e - F_j, 0)$. Now, it can be shown that F_{max} can be lowered by d_e if and only if a flow of size $\sum_{j=1}^{C} max(d_e + F_j - F_{max}, 0)$ can be realized from s to d, that is, if and only if all outgoing edges from source s have a flow equal to their capacity [57]. The maximum load F_{max} can be lowered by d_e units if and only if for each cluster j with $F_j > F_{max} - d_e$, F_j can be lowered by at least $F_j - F_{max} + d_e$ by appropriate reassignment of videos to other clusters, and for each cluster j with $F_j < F_{max} - d_e$, F_j is increased by at most $F_{max} - d_e - F_j$.

F_{max} being lower-bounded by $\frac{1}{C}$, a binary search strategy of complexity $O(log(C \cdot m))$ can be initiated from any arbitrary assignment (e.g., the greedy heuristic). In each iteration, listing all possible decrements in access frequencies for each cluster takes $O(C^2 s_C^2)$ time, and then solving a max-flow problem takes $O(C^3)$ time [92]. Hence, an optimal video allocation is found in $O((C^2 s_C^2 + C^3)\log(C \cdot s_C))$ time. □

APPENDIX C

The Average Stationary Approximation in $M(t)/D/s$ Systems

Stationary models used to approximate a nonstationary system can seriously underestimate delays. Although upper and lower bounds can be found using this approximation, a better strategy is to use the existing theoretical results for nonstationary arrival rates. In particular, we approximate the $M(t)/D/d_m.B$ queueing system corresponding to movie title m by using a combination of the numerical method in [46, 98] for $M(t)/M/s$ systems and the stationary approximation of $M/D/s$ from $M/M/s$ in [56].

To model nonstationary request arrival rates, we use the *average stationary approximation* (ASA), initially proposed for $M(t)/M/s$ queueing systems in [98]. Let $\widetilde{EW}(M_{\lambda_m(t)}/D/d_m.B)$ denote the waiting time of requests for movie title m averaged over a 24-period, then ASA stipulates:

$$\widetilde{EW}(M_{\lambda_m(t)}/D/d_m.B) \approx \frac{1}{T}\int_0^T \overline{EW}(M_{\overline{\lambda}_m(t)}/D/d_m.B)dt, \tag{C.1}$$

$$\overline{\lambda}_m(t) = \frac{1}{L}\int_{t-\kappa L}^{t} \lambda_m(s)ds, \tag{C.2}$$

where κ is a corrective positive constant (empirically determined), and $\overline{EW}(M_{\overline{\lambda}_m(t)}/D/d_m.B)$ is the average waiting time in receiving service corresponding to stationary arrivals at rate $\overline{\lambda}_m(t)$. In practice, a value of $\kappa = 1$ is recommended [98], in which case the arrival rate is averaged over a service interval. Numerical integration of Eq. (C.1) can be done with standard library routines. The choice of a sinusoidal arrival rate in Eq. (2.2) simplifies the integration of Eq. (C.2).

To ensure the existence of limiting distributions, it is shown in [46] that the following stability conditions have to hold for each title:

$$\begin{aligned} \overline{\rho}_m &= \frac{\overline{\lambda}_m}{d_m.B\mu} < 1 \\ RA &= \frac{A}{\overline{\lambda}} < 1, \end{aligned} \tag{C.3}$$

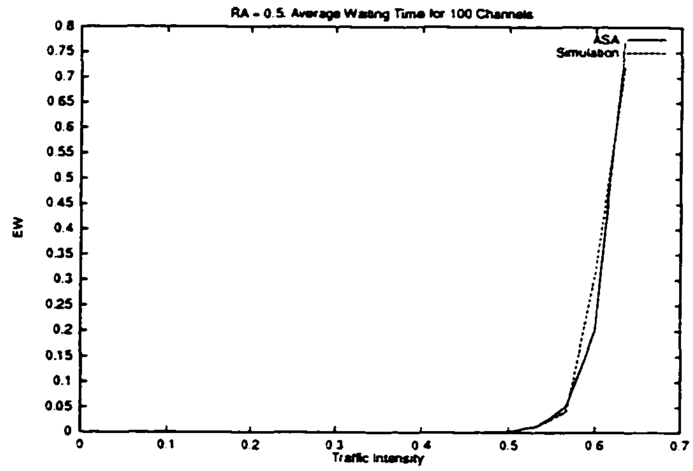


Figure C.1: ASA approximation for the average waiting time ($RA = 0.5$).

It is important to notice that due to both the stability conditions in Eq. (C.3) and the use of stationary systems in ASA, we are confined to systems with the maximum traffic intensity strictly less than 1:

$$\rho_m^{max} = \sup_m \frac{\lambda_m(t)}{d_m.B\mu} = \frac{\overline{\lambda}_m + p_m.A}{d_m.B\mu} < 1. \tag{C.4}$$

We ran simulations to compare the ASA approximation of the average waiting time in Eq. (C.1) with the exact value obtained from recursive simulation of the $M_{\overline{\lambda}_m(t)}/D/d_m.B$ system. For clarity of presentation, we chose to restrict the number of parameters and studied the variations of the average waiting time for movie title m, as a function of the average traffic intensity $\overline{\rho}_m$, for a constant value of $RA < 1$. The stability conditions can be written as:

$$\frac{\overline{\lambda}_m}{d_m.B\mu} < \frac{1}{1 + RA} \quad ; \quad \frac{A_m}{\overline{\lambda}_m} = RA. \tag{C.5}$$

Thus, for an arbitrary acceptable traffic intensity, we first determine the corresponding average arrival rate $\overline{\lambda}_m$ then the amplitude A_m.

The simulation results were plotted in Figure C.1 for 100 channels, with $RA = 0.5$ corresponding to a moderately nonstationary arrival rate. We obtained similar results for other channel capacities. The ASA approximation is shown to provide a precision that is likely to be acceptable for determining the concurrent access profile. We obtained similar results for higher values of RA, although with a slight loss of precision for high traffic intensities. In such cases, though, it was possible to adjust the value of κ to reach a desirable precision. In practice, we can tabulate the values of κ as a function of the traffic intensity and the number of servers for a given value of RA.

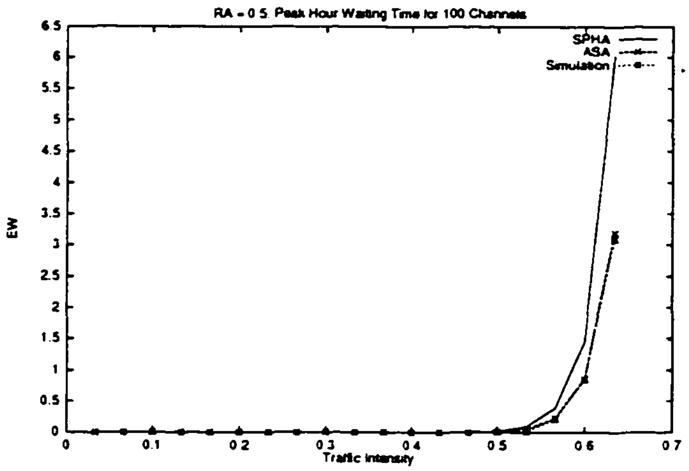


Figure C.2: ASA approximation for the peak-hour waiting time ($RA = 0.5$).

The ASA technique can also provide good performance estimation during peak-hour, and more generally during relatively short intervals of heavy congestion in D-VoD servers[1]. It is a standard practice to approximate peak-hour performance by using the stationary approximation based on the average arrival rate during "rush" hour (assuming a 24-hour cycle). In a sense, such approach, called *simple peak-hour approximation* (SPHA), is intuitively appealing since for small time durations, the average arrival rate can be viewed as constant. SPHA was evaluated in [48] for $M(t)/M/s$ systems, and it was shown to be very accurate with high service rates (e.g., in the hundreds of customers per hour) almost without regard for the value of other system parameters, and for cases more general than sinusoidal arrival rates and exponential service times. In our case, the SPHA approximation of the average waiting time in receiving service is obtained simply with the average arrival rate in Eq. (C.2) over the peak hour. In Figure C.2, we compare, for 100 channels and $RA = 0.5$ the peak-hour waiting times computed with (1) SPHA, (2) the ASA method on the peak hour, with appropriate value of κ (1.5 for 100 channels), and (3) the accurate waiting time obtained from simulation. The main result is that ASA performs better than SPHA for high traffic intensities, whereas SPHA is sufficiently precise for low traffic intensities. Thus, the ASA technique provides efficient performance estimation during relatively short intervals of highest congestion in D-VoD servers with periodic arrival rates.

[1] Though most of our discussion has focused on the peak hour, the choice of a one-hour interval is purely arbitrary. Our actual interest is in the more general issue of peak-period performance where the duration of the period should be defined based on actual observations. In practice, this is usually an hour. Our results can be generalized to any peak period of reasonably short duration.

APPENDIX D

Average Channel Rate in T-VoD

In order to determine the probabilities that a customer is in a particular state at any given time, the activity model presented Figure 2.14 can be transformed into a Markov chain with 5 states and transition rate from state i to state j inferred from $\mu_i, i = 0, \cdots, 5$ and $P_j, j = 0, \cdots, 6$. Alternatively, we choose to use the method in [21], based on queueing theory.

We model a channel serving one interacting customer as the closed Jackson network represented in Figure D.1. To each interaction i necessitating a different transmission rate or a different retrieval mechanism corresponds to an $M/M/\infty$ queueing system with average service time $\frac{1}{\mu_i}$. The exponential service time corresponds to the exponential duration of each interaction, assumed in the channel usage model. Fast-forward, rewind, and abort actions do not necessitate any data transmission; fast-forward and rewind are simply performed quasi-instantaneously by retrieving out-of-sequence blocks of video from the storage. From a queueing perspective, these actions are modeled by transitions. If no statistical difference is found between initial playback and random play/resume actions, the Jackson network can be further simplified by merging systems 1 and 2 and assigning to the resulting system the input probability $P_0 + P_1 + P_2$.

It is important to note that the closed network model is used as a tool to approximate the rate of a connection from the flow between different states. It should be based on a large number of observations of a large number of customers over a representative period of time. In reality, customers do not actually wait in line to get service for a particular interaction, nor are service times exponentially distributed or the number of waiting buffers infinite. Customers are assigned their own channel upon connection and keep it irrespectively of which state they are in.

Let's now calculate the portion of time spent by the customer in each queueing system

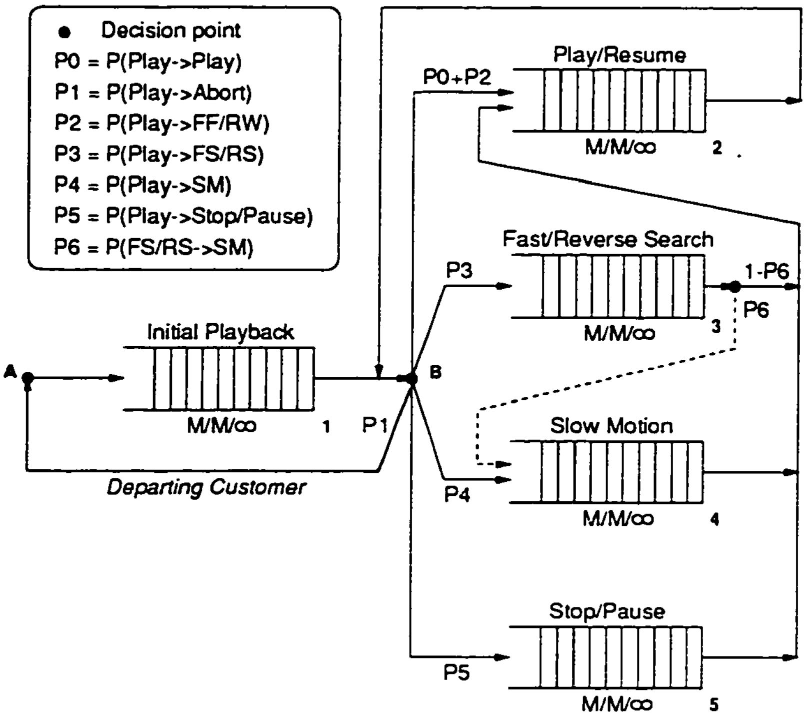


Figure D.1: Closed network model of a channel.

of the closed network in Figure D.1. These portions are given by $P(\overline{n}_j), j = 1, \cdots, 5$, probabilities of the customer being at system j, with

$$\overline{n}_j = [n_{j1}, \cdots, n_{j5}] \in \mathcal{N} = \{(1,0,0,0,0),(0,1,0,0,0),$$
$$(0,0,1,0,0),(0,0,0,1,0),(0,0,0,0,1)\}.$$

We denote the rate at each queueing system j by $\alpha_j = \alpha(1)\nu_j$, where $\alpha(1)$ is the sum of the arrival rates to all 5 systems when there is 1 customer in the closed network, and ν_j the relative portion of the arrival rate to system j. We have:

$$P(\overline{n}_j) = \chi(1,5)\prod_{i=1}^{5} p_i(n_{ji}), \tag{D.1}$$

where $\chi(1,5)$ is a constant which depends on the number of systems and the number of customers, and the $p_i(n_{ji})$ are the standard probability of i customers in the $M/M/\infty$ system j, given in [21]:

$$p_i(n_{ji}) = \rho_j^{n_{ji}}\frac{e^{-\rho_j}}{n_{ji}!}, \tag{D.2}$$

where $\rho_j = \frac{\alpha_j}{\mu_j}$. The normalization conditions $\sum_{\overline{n}_j\in\mathcal{N}} P(\overline{n}_j) = 1$ and $n_{ji} = 1$ for $i = j$, 0 otherwise, leads to:

$$\chi(1,5) = \frac{\exp\left(\sum_{j=1}^{5}\rho_j\right)}{\sum_{j=1}^{5}\rho_j}. \tag{D.3}$$

We obtain the portion of time spent by the customer in each interaction system:

$$P(\overline{n}_j) = \frac{\frac{\nu_j}{\mu_j}}{\sum_{i=1}^{5} \frac{\nu_i}{\mu_i}}. \tag{D.4}$$

By flow conservation, $\overline{\nu} = [\nu_1, \nu_2, \nu_3, \nu_4, \nu_5]$ is solution of $\overline{\nu} P = \overline{\nu}$, where P is the one-step transition matrix given by:

$$P = \begin{bmatrix} P_1 & P_0 + P_2 & P_3 & P_4 & P_5 \\ P_1 & P_0 + P_2 & P_3 & P_4 & P_5 \\ 0 & 1 - P_6 & 0 & P_6 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

After calculations, we obtain:

$$\begin{aligned} \nu_2 &= \frac{1 - P_1}{P_1}\nu_1 \\ \nu_3 &= \frac{P_3}{P_1}\nu_1 \\ \nu_4 &= \frac{P_4 + P_3 P_6}{P_1}\nu_1 \\ \nu_5 &= \frac{P_5}{P_1}\nu_1 \end{aligned}$$

where ν_1 can be found by $\sum_{j=1}^{5} \nu_j = 1$. Finally, replacing $\nu_j, j = 1, \cdots, 5$ by their values given in Eq. (D.4) results in Eq. (2.4).

APPENDIX E

Analysis of the $M_t/M/\infty$ Patience Queue

Here we calculate the average number of customers waiting for service at the end of a reservation period, which corresponds to the phase offset before receiving service. The main $M_t/G/\infty$ result, of which $M_t/M/\infty$ is a special case, is thanks to Palm and Khintchine. This result, presented in [36], states that the number of busy servers at time t, which is, in the patience queue, the actual number of customers waiting for service, has a Poisson distribution, and is therefore fully specified by its average. In our case of exponential patience (or service rate), the *average* number of busy servers in $M_t/M/\infty$ can be found from the following differential equation:

$$\begin{aligned} \dot{L}_{M_t/M/\infty}(t,t_0) + \alpha L_{M_t/M/\infty}(t,t_0) &= \lambda_m(t) && \text{(E.1)} \\ L_{M_t/M/\infty}(t,t_0) &= 0 \qquad t \le t_0. && \text{(E.2)} \end{aligned}$$

where

$$\dot{L}_{M_t/M/\infty}(t,t_0) = \frac{d}{dt}\left(L_{M_t/M/\infty}(t,t_0)\right).$$

We added Eq. (E.2) to represent the initial conditions of the $M_t/M/\infty$ system in case of NVoD, which states that the patience queue restarts empty at the beginning of each phase offset. After some calculations, we find that for the sinusoidal arrival rate of Section 3.3.2, the general solution of Eq. (E.1) is given by:

$$L_{M_t/M/\infty}(t,t_0) = \frac{\overline{\lambda}_m}{\alpha} + \left(\frac{A_m}{\alpha+\gamma^2}\right)\left\{\sin(\gamma t) - \frac{\gamma}{\alpha}\cos(\gamma t)\right\} + \left(\frac{\gamma}{\alpha}\cdot\frac{A_m}{\alpha+\gamma^2} - \frac{\overline{\lambda}_m}{\alpha}\right)e^{-\alpha(t-t_0)}. \tag{E.3}$$

APPENDIX F

Influence of System Parameters on VCR Functionality

In this appendix, we assume that a particular batching policy has been chosen, following the guidelines of Sections 5.4.2 and 5.4.3 of Chapter 5, and we focus on the VCR functionality experienced by customers after admission.

F.1 CPE-Buffer Size

The size of the CPE buffer relative to the duration of VCR actions will determine the fraction of CPE-actions, and, possibly, that of discontinuous and I-actions. So, we considered CPE buffers holding 0.5, 1, and 2 minutes' worth of video data, for interactions of average duration 1 minute. Figure F.1 plots the simulation results obtained for NVoD — similar results were obtained for other batching policies — infinitely patient customers, and for (B, I) = (200,0) and (150,50). For low traffic intensities, increasing the fraction of I-channels, for instance, from 0% to 25%, may be a cheaper alternative to reduce the

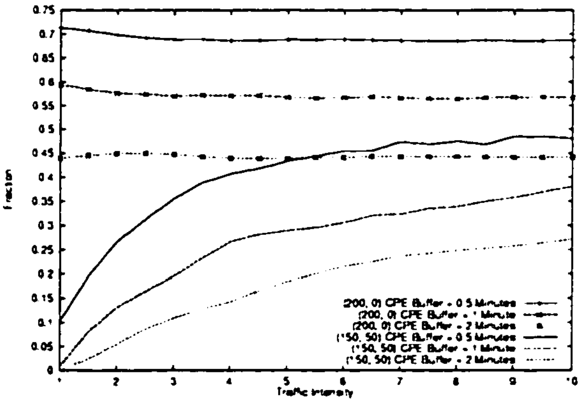


Figure F.1: Average fraction of blocked VCR actions for various CPE-buffer sizes.

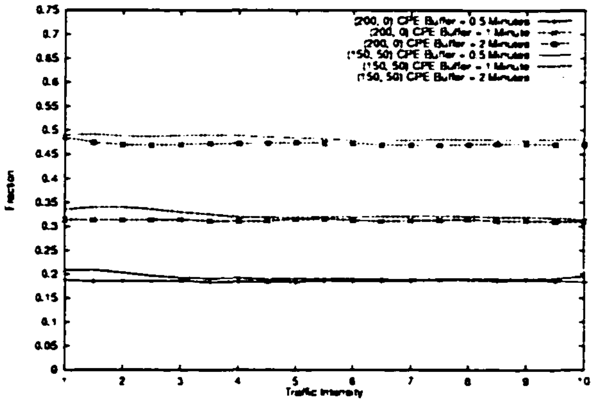


Figure F.2: Average fraction of CPE-actions for various CPE-buffer sizes.

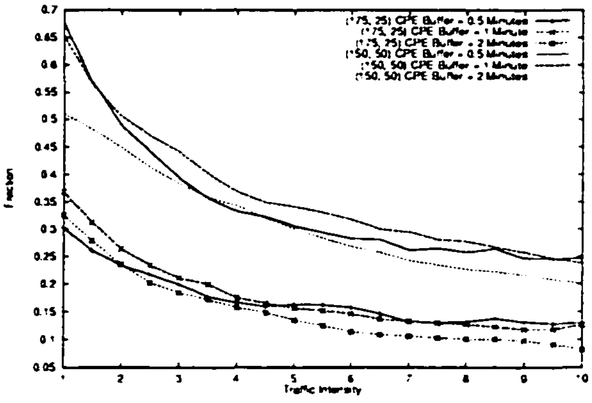


Figure F.3: Average fraction of I-actions for various CPE-buffer sizes.

fraction of blocked actions. As the traffic intensity increases, however, the I-channel capacity becomes saturated and a large CPE buffer is preferable to maintain scalability. Similar results were obtained for the fraction of discontinuous actions. To some extent, the affordability of the CPE is an alternative form of pricing in VoD systems, since the more users are willing to pay for additional buffer space, the closer they get to truly interactive VoD service. As illustrated in Figures F.2 and F.3, changes in the fraction of I-channels and in CPE-buffer size have almost independent effects on the partition of VCR actions. Surprisingly, increasing the CPE-buffer size for a given partition (B, I) will only marginally affects the number of I-actions, even though we found that the probability of merge success increases almost in proportion to the increase in buffer size. Correspondingly, increasing the fraction of I-channels for a fixed CPE-buffer size will not affect the fraction of CPE-actions.

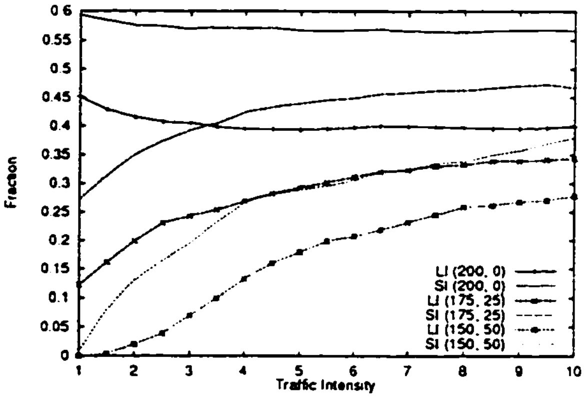


Figure F.4: Average fraction of blocked VCR actions for various durations.

F.2 Channel Capacity

To test the sensitivity of continuous VCR functionality to the channel capacity, we consider two configurations, *SI* and *LI*, such that the ratio of CPE-buffer size to average duration of VCR actions is kept constant: (1) a CPE buffer of size 1 minute's worth of video, for an average duration of VCR actions set to 1 minute; (2) a CPE buffer of size 5 minutes' worth of video, for an average duration of VCR actions set to 5 minutes. In a sense, the average phase offset for each movie title, as seen by an individual customer, appears scaled down 5 times in the second configuration. We are thus able to simulate a 1,000-channel capacity with the same B- and I- channel breakdown. Figures F.4 and F.5 represent the breakdown between blocked and discontinuous actions for both configuration *SI* and *LI* in 3 cases of channel partition, (B, I) = (200, 0), (175, 25), and (150, 50). If we consider the sum of blocked and discontinuous actions, there is no major difference in customers' QoS between the two configurations if the fraction of I-channels is kept the same. These results were consistent with similar fractions observed in each configuration for CPE- and I- actions.

The difference in fractions of discontinuous actions between *SI* and *LI* configurations can be explained similarly to the analysis of the greater fraction of discontinuous actions in NVoD T-OPT in Section 5.4.3 of Chapter 5. As can be seen from Figure F.6, in the *LI* configuration, the relative discrepancy experienced by customers in discontinuous actions is reduced since the average phase offset is small relative to the duration of VCR actions. Recall that a lower discrepancy tends to shorten program subjective duration experienced by customers in the *LI* configuration. The absolute durations of pause and stop actions, and the average jump sizes in fast forward and rewind, being the same, the relative frac-

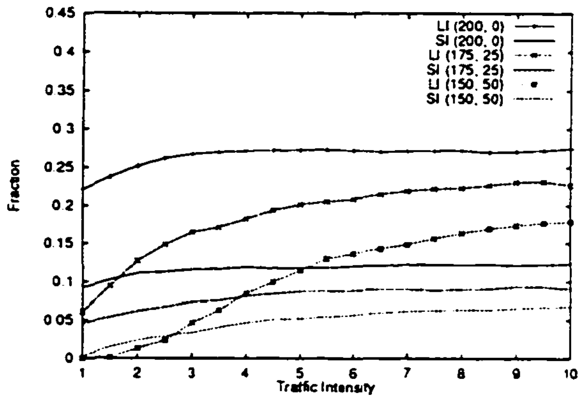


Figure F.5: Average fraction of discontinuous actions for various durations.

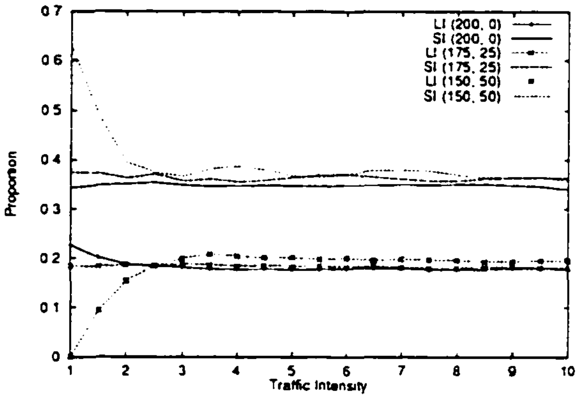


Figure F.6: Average discrepancy of discontinuous actions for various durations.

tion of discontinuous actions is greater in the *LI* configuration. Consequently, support for discontinuous operations is more effective with frequently-requested movies, for which more channels are allocated.

We can take advantage of this last feature to provide a more equitable service to customers viewing less popular movies as follows. First, each movie title can be considered as either *hot* or *cold*, depending on the number of B-channels allocated to that movie, or, for instance, depending on whether the average phase offset of that movie is smaller or greater than the CPE-buffer size. The VoD server can then allocate more I-channels to cold movies, hence providing customers viewing these movies with a better support for VCR actions. This approach, albeit counter-intuitive, can effectively prevent customers viewing the most popular movies from saturating the I-channel capacity by unnecessarily using I-channels for pause, stop, fast forward and rewind actions; these same VCR actions could have been served discontinuously with very low, and therefore acceptable, discrepancy. We

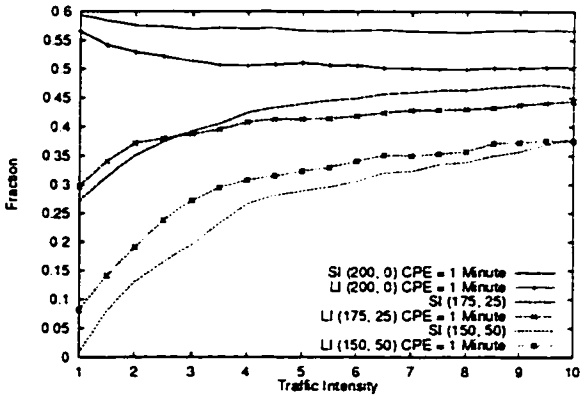


Figure F.7: Average fraction of blocked VCR actions for various durations.

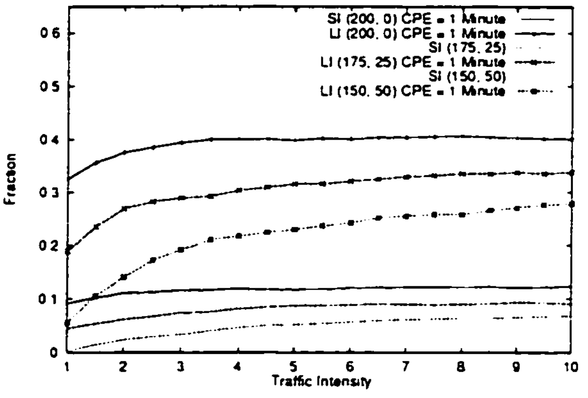


Figure F.8: Average fraction of discontinuous actions for various durations.

leave as future work the issue of determining the breakdown between hot and cold movies, and, correspondingly, an optimal partition of the I-channels. It is worth noticing that channel allocation can be further refined by using QVoD-enhanced NVoD batching for cold movies, and NVoD for hot movies. In the former case, customers for less popular movies are not forced to wait unnecessarily for half a phase offset, while customers requesting popular movies can take advantage of the small phase offsets corresponding to their movies.

F.3 Interactive Behavior

We next study the effect of variations in duration of VCR actions. For a CPE-buffer size set to 1 minute's worth of video, we consider two average durations, 1 minute (referred to as *SI*) and 5 minutes (*LI* configuration), and in each case, three partitions of a fixed 200-channel capacity. According to our previous numerical study with various CPE-buffer sizes,

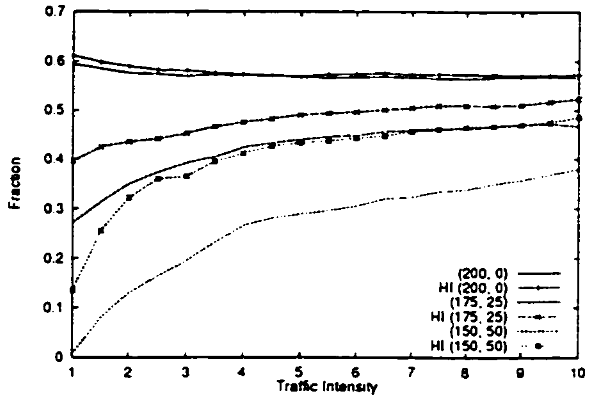


Figure F.9: Average fraction of blocked VCR actions for various interactivity levels.

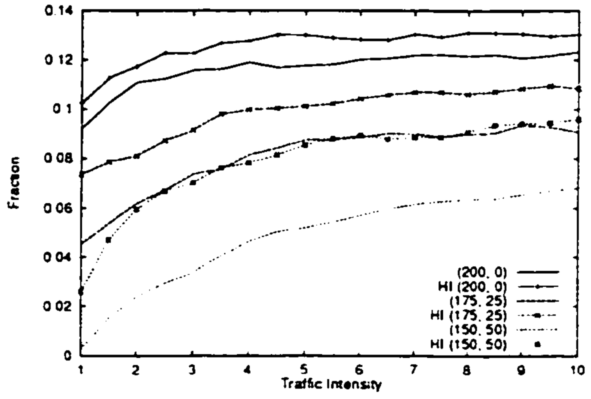


Figure F.10: Average fraction of discontinuous actions for various interactivity levels.

we can expect the fraction of CPE-actions to be lower in the LI configuration, while the fraction of I-actions is likely to be the same for both SI and LI configurations. Figures F.7 and F.8 confirm that the loss in CPE-actions is mostly translated into a higher fraction of discontinuous actions of reduced discrepancy (not represented here), whereas the fraction of blocked actions remains relatively unaffected. Support for discontinuous operations is therefore effective in graceful QoS degradation when the duration of VCR actions is not precisely known, or increases randomly.

In Figure F.9 and F.10, we test the sensitivity of customers' QoS to variations in the level of interactivity. We distinguish between normally (used by default) and highly interactive customers (referred to as HI) for whom the transition probability from play mode to any other state is doubled to 50%, as indicated in Table 5.1. A higher level of interactivity generates more demand on the CPE buffer and on the pool of I-channels. The fraction of I-actions for highly interactive customers is decreased by 50% in both $(B, I) = (175, 25)$

and $(B, I) = (150, 50)$ partitions, whereas the fraction of CPE-actions generally decreased from 32% to 30%. Consequently, the CPE buffer is a robust support for continuous service of VCR actions when customers' level of interactivity varies. Also, the loss in QoS depicted Figure F.9 and F.10, is mainly due to blocked actions and increases with the fraction of I-channels, from relatively minor when no I-channel is provided to approximately 30% for $(B, I) = (150, 50)$. The increase in discontinuous actions confirms the effective support for graceful QoS degradation provided by discontinuous operations.

APPENDIX G

List of Acronyms

ADSL: Asymmetric Digital Subscriber Line

APO: Average Phase Offset

ASA: Average Stationary Approximation

ATM: Asynchronous Transfer Mode

BAC: Burst Absorption Capability

BB: Backward Bias

B-Channel: Broadcast Channel

CAP EW-PROP: Concurrent Access Profile obtained by allocating the number of channels per movie in proportion to popularities

CATV: Community Access TV

CBR: Constant Bit Rate

CGS: Coarse-Grained Striping

CP: Completely-Partitioned

CPE: Customers' Premise Equipment

CS: Completely-Shared

DPSR: Dynamic Policy of Segment Replication

DUP: Duplication

DVD: Digital Versatile Disc

D-VoD: Deterministic Video-on-Demand

EPPV: Enhanced Pay-Per-View

FA: Folding Algorithm

FB: Forward Bias

FCFS: First-Come First-Served

FF: Fast Forward

FGS: Fine-Grained Striping

FS: Fast Search

FSA: Fiber Service Areas

FTTC: Fiber-To-The-Curb

FTTH: Fiber-To-The-Home

GOP: Group of Pictures

HCGS: Horizontal Coarse-Grained Striping

HFC: Hybrid-Fiber-Coax

I-Channel: Interaction Channel

LCFS: Last-Come First-Served

LI: Long Interactions

LSCR: Latency During Single-Cluster Reconfigurations

MFQL: Maximum Factored Queue Length

MPEG: Motion Picture Expert Group

MQL: Maximum Queue Length

MTTF: Mean Time to Failure

NB: Neutral Bias

NEC: Normalized Effective Cost

NVI: Not Very Interactive (Customers)

NVoD: Near Video-on-Demand

NVoD-h: Heterogeneous Near Video-on-Demand

NVoD EW-OPT (or EW-OPT in Chapter 3): NVoD server in which the channel capacity is partitioned among movie titles so the average phase offset is minimized; simply called NVoD in the case of infinitely-patient customers (cf. Chapter 5)

NVoD T-OPT (or T-OPT in Chapter 3): NVoD server in which the channel capacity is partitioned among movie titles so the server throughput is maximized (or the defection rate minimized)

NVoD T-PROP (or T-PROP in Chapter 3): NVoD server in which the channel capacity is partitioned among movie titles in proportion to movie popularities

NVoD T-SQRT (or T-SQRT in Chapter 3): NVoD server in which the channel capacity is partitioned among movie titles in proportion to the square root of movie popularities

NVoD-VCR: Work-Conserving Near Video-on-Demand

OCAP: Optimal Concurrent Access Profile

OCAP EW-OPT: Optimal Concurrent Access Profile, obtained by minimizing the average admission latency

OD: On-Demand batching

OMPD: One-Movie-Per-Disk

PBPB: Permutation-Based Pyramid Broadcasting

PMSP: Periodic Maintenance Scheduling Problem

QoS: Quality-of-Service

QVoD: Quasi Video-on-Demand (or threshold-based NVoD)

QVoD-h (or QVoD in Chapter 5): Heterogeneous Quasi Video-on-Demand

QVoD-Enhanced NVoD: QVoD system used over a partition of the channel capacity among movie titles initially determined for NVoD

QNVoD: QVoD-Enhanced NVoD

QVoD-enhanced NVoD EW-OPT (or QNVoD EW-OPT in Chapter 4): QVoD-Enhanced NVoD server in which the channel capacity is partitioned among movie titles so the average phase offset in the corresponding NVoD server is minimized: simply called QVoD-Enhanced NVoD in the case of infinitely-patient customers (cf. Chapter 5)

QVoD-enhanced NVoD T-OPT (or QNVoD T-OPT in Chapter 4): QVoD-Enhanced NVoD server in which the channel capacity is partitioned among movie titles so the server throughput in the corresponding NVoD server is maximized (or the defection rate minimized)

QVoD-enhanced NVoD T-PROP: QVoD-Enhanced NVoD server in which the channel capacity is partitioned among movie titles in proportion to movie popularities

QVoD-enhanced NVoD T-SQRT: QVoD-Enhanced NVoD server in which the channel capacity is partitioned among movie titles in proportion to the square root of movie popularities

RC: Rate Control

RS: Reverse Search

RW: Rewind

SA: Stationary Approximation

SAM: Split-and-Merge protocol

SI: Short Interactions

SM: Slow Motion Factor

SP: Speedup Factor

SPHA: Simple Peak-Hour Approximation

STB: Set-Top Box

TAPMO: Threshold-Activated Preventive Merge Operation

T-VoD: True Video-on-Demand

VoD: Video-on-Demand

VBR: Variable Bit Rate

VCGS: Vertical Coarse-Grained Striping

VI (or HI): Very (or Highly) Interactive (Customers)

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] E. L. Abram-Profeta and K. G. Shin, "Comparative Study of Scalable Batching Policies in Disk-Array-Based Deterministic Video-on-Demand Servers," *Submitted for publication*, 1998.

[2] E. L. Abram-Profeta and K. G. Shin, "Scalable Batching Policies in Disk-Array-Based Video-on-Demand Servers," *Submitted for publication*, 1998.

[3] E. L. Abram-Profeta and K. G. Shin, "Active Buffer Management for Unrestricted VCR Capability in Multicast Video-on-Demand Systems," *Submitted for publication*, 1998.

[4] E. L. Abram-Profeta and K. G. Shin, "A Practical Approach to Resource Allocation in Video-on-Demand Servers," *Submitted for publication*, 1998.

[5] E. L. Abram-Profeta and K. G. Shin, "Providing Unrestricted VCR Functions in Multicast Video-on-Demand Servers," *IEEE International Conference on Multimedia Computing and Systems*, June-July 1998.

[6] E. L. Abram-Profeta and K. G. Shin, "Scheduling Video Programs in Near Video-on-Demand Systems (Extended Version)," *Submitted for publication*, 1997.

[7] E. L. Abram-Profeta and K. G. Shin, "Scheduling Video Programs in Near Video-on-Demand Systems," *Proc. ACM Multimedia'97*, pp. 359-369, November 1997.

[8] C. C. Aggarwal, J. L. Wolf and P. S. Yu, "A Permutation-Based Pyramid Broadcasting Scheme for Video-on-Demand Systems," *IEEE International Conference on Multimedia Computing and Systems (Multimedia'96)*, pp. 118-126, 1996.

[9] C. C. Aggarwal, J. L. Wolf and P. S. Yu, "Adaptive Piggybacking Schemes for Video-on-Demand Systems," *IBM Research Report RC 20635*, Yorktown Heights, NY, November 1996.

[10] C. C. Aggarwal, J. L. Wolf and P. S. Yu, "On Optimal Batching Policies for Video-on-Demand Storage Servers," *Proc. ACM Multimedia'96*, pp. 253-258, 1996.

[11] K. C. Almeroth, A. Dan, D. Sitaram, and W. H. Tetzlaff, "Long Term Resource Allocation in Video Delivery Systems," *Proc. IEEE INFOCOM'97*, April 1997.

[12] K. C. Almeroth and M. H. Ammar, "The Use of Multicast Delivery to Provide a Scalable and Interactive Video-on-Demand Service," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 6, pp. 1110-1122, August 1996.

[13] K. C. Almeroth and M. H. Ammar, "On the Performance of a Multicast Delivery Video-On-Demand Service with Discontinuous VCR Actions," *Proc. of ICC'95*, pp. 1631-1635, June 1995.

[14] J. B. Bailey, "Opening the Set-Top Box Market," *Working Paper*, Research Program on Communications Policy, Massachusetts Institute of Technology, February 1995 (may be found at http://rpcp.mit.edu/Pubs/).

[15] S. A. Barnett and G. J. Anido, "A Cost Comparison of Distributed and Centralized Approaches to Video-on-Demand," *IEEE Journal on Selected Areas in Communications*, 14, 6, pp. 1173-1183, August 1996.

[16] S. Baruah, L. Rosier, I. Tulchinsky, and D. Varvel, "The Complexity of Periodic Maintenance," *Proc. of the 1990 Intl. Computer Symp.*, Taiwan, 1990.

[17] D. Bertsekas, R. Gallagher, *Data Networks*, Prentice Hall, Inc., New Jersey, 1987.

[18] G. Bianchi, R. Melen, "Non stationary request distribution in Video on Demand Networks," *Proc. IEEE INFOCOM'97*, April 1997.

[19] E. W. Biersack, C. Bernhardt, "A Fault Tolerant Video Server Using Combined Raid 5 and Mirroring," *Proc. MMNC'97, SPIE Vol. 3020*, pp. 106-117, 1997.

[20] C. C. Bisdikian and B. V. Patel, "Issues on Movie Allocation in Distributed Video-on-Demand Systems," *Proc. of ICC'95*, pp. 250-255, June 1995.

[21] B. D. Bunday, *An Introduction to Queueing Theory*, Arnold, 1996.

[22] M. J. Carillo, "Extensions of Palm's Theorem: A Review," *Management Science*, Vol. 37, No. 6, pp. 739-744, June 1991.

[23] S. Chakravarthy, A. Sule Alfa, "A Multiserver Queue with Markovian Arrivals and Group Services with Thresholds," *Naval Research Logistics*, Vol. 40, pp. 811-827, 1993.

[24] E. Chang and A. Zakhor, "Cost Analyses for VBR Video Servers," *Proc. MMNC'96, SPIE Vol. 2667*, pp. 381-397, 1996.

[25] M.-S. Chen, D. D. Kandlur, and P. Yu, "Support for Fully Interactive Playout in a Disk-Array-Based Video Server," *Proc. ACM Multimedia'94*, pp. 391-398, October 1994.

[26] A. L. Chervenak, D. A. Patterson, R. H. Katz, "Choosing the Best Storage System for Video Services," *Proc. ACM Multimedia'95*, pp. 109-119, 1995.

[27] S. Christian, "Software Design Issues for Digital Set-Top Box Applications," *Multimedia Systems Design*, Premiere Issue, Miller Freeman, Inc., December 1997 (may be found at http://www.msdmag.com/97/11settop.htm).

[28] T. S. Chua, J. Li, B. C. Ooi, K.-L. Tan, "Disk Striping Strategies for Large Video-on-Demand Servers," *Proc. ACM Multimedia'96*, 1996.

[29] M. E. Crovella and A. Bestavros. "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurements and modeling of Computer Systems*, pp. 160-169, May 1996.

[30] A. Dan, D. Sitaram, "A Generalized Interval Caching Policy for Mixed Interactive and Long Video Workloads," *Proc. MMNC'96, SPIE Vol. 2667*, pp. 344-351, 1996.

[31] A. Dan, P. Shahabuddin, D. Siraram, and D. Towsley, "Channel Allocation under Batching and VCR Control in Video-on-Demand Systems," *Journal of Parallel and Distributed Computing*, **30**, pp. 168-179, 1995.

[32] A. Dan, M. Kienzle, D. Sitaram, "A dynamic policy of segment replication for load-balancing in video-on-demand servers," *Multimedia Systems*, **3**, 3, pp. 93-103, July 1995.

[33] A. Dan, D. M. Dias, R. Mukherjee, D. Sitaram and R. Tewari. "Buffering and Caching in Large-Scale Video Servers," *IEEE COMPCON'95*, pp. 217-224, March 1995.

[34] D. Deloddere, W. Verbiest, and H. Verhille. "Interactive Video On Demand," *IEEE Communications Magazine*, Vol. 32, no. 5, pp. 82-88, May 1994.

[35] J. K. Dey-Sircar, J. D. Salehi, J. F. Kurose, D. Towsley, "Providing VCR Capabilities in Large-Scale Video Servers," *Proc. ACM Multimedia'94*, pp. 25-32, October 1994.

[36] S. G. Eick, W. A. Massey, W. Whitt, "$M_t/G/\infty$ Queues with Sinusoidal Arrival Rates," *Management Science*, Vol. 39, No. 2, pp. 241-252, February 1993.

[37] W.-C. Feng, F. Jahanian, and S. Sechrest, "Providing VCR Functionality in a Constant Quality Video-on-Demand Transportation Service", *IEEE International Conference on Multimedia Computing and Systems (Multimedia'96)*, pp. 127-135, 1996.

[38] R. Flynn and W. Tezlaff, "Disk Striping and Block Replication Algorithms for Video File Servers," *Proc. ACM Multimedia'96*, pp. 590-597, 1996.

[39] B. Fox, "Discrete optimization via marginal analysis," *Management Science*, **13**, pp. 210-216, November 1966.

[40] B. Furht, D. Kalra, and A. A. Rodriguez, "Interactive Television Systems," *Multimedia Tools and Applications*, pp. 235-277, Kluwer Academic Publications, Inc., 1997.

[41] M. Garofalakis, B. Özden, A. Silberschatz, "On Periodic Resource Scheduling for Continuous Media Databases," *RIDE*, February 1998.

[42] M. Garofalakis, B. Özden, A. Silberschatz, "Resource Scheduling in Enhanced Pay-Per-View Continuous Media Databases," *International Conference on Very Large Databases*, September 1997.

[43] A. D. Gelman, L. S. Smoot, "An Architecture for Interactive Applications," *Proc. of ICC'93*, pp. 848-852, May 1993.

[44] A. D. Gelman, S. Halfin, "Analysis of Resource Sharing Information Providing Services," *Proc. IEEE GLOBECOM'90*, December 1990, pp. 312-316.

[45] H. Ghafir and H. Chadwick, "Multimedia Servers - Design and Performance," *IEEE GLOBECOM'94*, pp. 886-890, December 1994.

[46] L. Green and P. Kolesar, "The Pointwise Stationary Approximation for Queues with Nonstationary Arrivals," *Management Science*, Vol. 37, No. 3, pp. 84-97, January 1991.

[47] L. Green, P. Kolesar and A. Svoronos, "Some Effects of Nonstationarity on Multiserver Markovian Queueing Systems," *Operations Research*, Vol. 39, No. 3, pp. 502-511, May-June 1991.

[48] L. Green, P. Kolesar, "On the Accuracy of the Simple Peak Hour Approximation for Markovian Queues," *Management Science*, Vol. 41, No. 8, pp. 1353-1369, August 1995.

[49] D. P. Heyman, M. J. Sobel, *Stochastic Models in Operations Research*, McGraw-Hill, 1982.

[50] I. Hsu and J. Walrand, "Quick Detection of Changes in Traffic Statistics: Application to Variable Rate Compression," *32nd Annual Allerton Conference on Communications, Control, and Computing*, Urbana-Champaign, IL, September 1994.

[51] T. Ibaraki, and N. Katoh, *Resource Allocation Problems*, MIT Press, Cambridge, MA, 1988.

[52] C. N. Judice, E. J. Addeo, M. I. Eiger, H. L. Lemberg, "Video on Demand: A Wideband Service or Myth?," *Proc. of ICC'86*, pp. 1735-1739, June 1986.

[53] K. Keeton, R. H. Katz, "Evaluating video layout strategies for a high-performance storage server," *Multimedia Systems*, **3**, 2, pp. 43-52, 1995.

[54] L. Kleinrock, *Queueing Systems, Volume 1: Theory*, John Wiley & Sons, 1975.

[55] M. G. Kienzle, A. Dan, D. Sitaram, W. Tetzlaff, "Effect of Video Server Topology on Contingency Capacity Requirements," *Proc. MMNC'96, SPIE Vol. 2667*, pp. 320-327, 1996.

[56] T. Kimura, "Refining Cosmetatos' Approximation for the Mean Waiting Time in the $M/D/s$ Queue," *Journal of the Operational Research Society*, **42**, 7, pp. 595-603, 1991.

[57] J. Korst, "Random Duplicated Assignment: An Alternative to Striping in Video Servers," *Proc. ACM Multimedia'97*, pp. 219-226, 1997.

[58] R. Krishnan, D. Venkatesh, T. D. C. Little, "A Failure and Overload Tolerance Mechanism for Continuous Media Servers," *Proc. ACM Multimedia'97*, 1997.

[59] S. Lederman, "Congestion Model for Subscriber Line Modules," *Proc. IEEE GLOBECOM'85*, pp. 395-401, 1985.

[60] S. Lederman, "Video-on-Demand - A Traffic Model and GOS Technique," *Proc. IEEE GLOBECOM'86*, pp. 676-683, 1986.

[61] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Communications of the ACM*, Vol. 34, no. 4, pp. 395-313, April 1991.

[62] Y.-W. Leung, and T.-S. Yum, "A Modular Multirate Video Distribution System - Design and Dimensioning," IEEE/ACM Transactions on Networking, Vol. 2, No. 6, pp. 549-557, 1994.

[63] V. O. K. Li, W. Liao, X. Qiu, and E. W. M. Wong, "Performance Model of Interactive Video-on-Demand Systems," *IEEE Journal on Selected Areas in Communications*, **14**, 6, pp. 1099-1109, August 1996.

[64] W. Liao and V. O. K. Li, "The Split and Merge (SAM) Protocol for Interactive Video-on-Demand Systems," *Proc. IEEE INFOCOM'97*, April 1997.

[65] T. D. C. Little, D. Venkatesh, "Popularity-based assignment of movies to storage devices in a video-on-demand system," *Multimedia Systems* (Springer, ACM Press, Ed.), **2**, 6, pp. 280-287, 1995.

[66] L. Liu, B. R. K. Kashyap and J. G. C. Templeton, "The Service System $M/M^R/\infty$ with Impatient Customers," *Queueing Systems*, No. 2, pp. 363-372, 1987.

[67] J. M. McManus and K. W. Ross, "Video-on-Demand over ATM: Constant-Rate Transmission and Transport," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 6, pp. 1087-1098, August 1996.

[68] G. Meempat and M. K. Sundareshan, "Optimal Channel Allocation Policies for Access Control of Circuit-Switched Traffic in ISDN Environments," *IEEE Transactions on Communications*, 41, 2, pp. 338-350, February 1993.

[69] M. F. Neuts, R. Nadarajan, "A Multiserver Queue with Thresholds for the Acceptance of Customers into Service," *Operations Research*, Vol. 30, No. 5, pp. 948-960, September-October 1982.

[70] K. W. Ng and K. H. Yeung, "Analysis on Disk Scheduling for Special User Functions," *Proc. ACM Multimedia'96*, pp. 608-611, 1996.

[71] K. Nishikawa, H. Egawa, O. Kawai, and Y. Inamoto, "High-Performance VoD Server AIMS," *Proc. IEEE GLOBECOM'95*, pp. 795-798, November 1995.

[72] J.-P. Nussbaumer, B. V. Patel, F. Schaffa, and J. P. G. Sterbenz, "Networking Requirements for interactive Video on Demand," *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 5, pp. 779-787, June 1995.

[73] F. Schaffa and J.-P. Nussbaumer, "On Bandwidth and Storage Tradeoffs in Multimedia Distribution Networks," *Proc. of IEEE INFOCOM'95*, pp. 1020-1026, April 1995.

[74] J.-P. Nussbaumer, B. V. Patel and F. Schaffa, "Multimedia Delivery on Demand: Capacity Analysis and Implications," *Proc. 19th Conference on Local Computer Networks*, pp. 380-386, October 1994.

[75] B. Özden, R. Rastogi, and A. Silberschatz, "Periodic Retrieval of Videos from Disk Arrays," *IEEE International Conference on Data Engineering*, April 1997.

[76] B. Özden, R. Rastogi, A. Silberschatz, "Fault-tolerant Architectures for Continuous Media Servers," *Proceedings of SIGMOD Conference*, pp. 79-90, 1996.

[77] B. Özden, R. Rastogi, A. Silberschatz, "Disk Striping in Video Server Environment," *Proc. IEEE International Conference on Multimedia Computing and Systems*, pp. 580-589, 1996.

[78] B. Özden, R. Rastogi, A. Silberschatz, and C. Martin, "Demand Paging for Video on Demand Servers," *Proc. IEEE International Conference on Multimedia Computing and Systems*, May 1995.

[79] B. Özden, A. Biliris, R. Rastogi, and A. Silberschatz, "A Disk-Based Storage Architecture for Movie on Demand Servers," *Information Systems*, Special Issue on Multimedia Information, Vol. 20, No. 6, pp. 465-482, 1995.

[80] B. Özden, A. Biliris, R. Rastogi, and A. Silberschatz, "A Low-Cost Storage Server for Movie on Demand Databases," *Proc. of the 20th International Conference on Very Large Databases*, pp. 594-605, September 1994.

[81] P. Pancha, M. El Zarki, "Leaky Bucket Access Control for VBR MPEG Video," *Proc. of IEEE INFOCOM'95*, pp. 796-803, April 1995.

[82] P. Pancha and M. El Zarki, "MPEG Coding for Variable Bit Rate Video Transmission," *IEEE Communications Magazine*, Vol. 32, no. 5, pp. 54-66, May 1994.

[83] P. Pancha and M. El Zarki, "Bandwidth-Allocation Schemes for Variable-Bit-Rate MPEG Sources in ATM Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 3, no. 3, pp. 190-198, June 1993.

[84] P. Pancha and M. El Zarki, "A look at the MPEG video coding standard for variable bit rate video transmission," *Proc. of IEEE INFOCOM'92*, pp. 85-93, 1992.

[85] W. B. Powell, "Waiting-Time Distributions for Bulk Arrival, Bulk Service Queues with Vehicle-Holding and Cancellation Strategies," *Naval Research Logistics*, Vol. 34, pp. 207-227, 1987.

[86] R. Sathiya Moorthi and V. Ganesan, "A Multichannel Queueing System with Hyper-Poisson Arrivals and Service in Batches," *Pure and Applied Mathematika Sciences*, Vol. 23, No. 1-2, March 1991.

[87] H. Shachnai and P. S. Yu, "The Role of Wait Tolerance in Effective Batching: A Paradigm for Multimedia Scheduling Schemes," *Technical Report RC 20038(88607)*, IBM Research Division, April 1995.

[88] P. J. Shenoy and H. M. Vin, "Efficient Support for Scan Operations in Video Servers," *Proc. ACM Multimedia'95*, pp. 131-140, 1995.

[89] S. H. Sim and J. G. C. Templeton, "Further Results for the $M/M(a,\infty)/N$ Batch-Service System," *Queueing Systems*, No. 6, pp. 177-286, 1990.

[90] A. S. Tanenbaum, *Computer Networks*, Third Edition, Prentice Hall, Inc., New Jersey, pp.744-757, 1996.

[91] N. Terada, H. Ishii, T. Tachi, Y. Okumura, H. Kotera, "An MPEG2-Based Digital CATV and VOD System using ATM-PON Architecture," *Proc. ACM Multimedia'96*, pp. 522-531, 1996.

[92] J. Van Leeuwen, "Graph algorithms," *Handbook of Theoretical Computer Science*, Vol. A, Algorithms and Complexity, pp. 525-631, Elsevier/MIT Press.

[93] J. Vandenameele, G. Van der Plas, C. Sierens, O. Nielsen, S. Graugaard-Jensen, "How to upgrade CATV networks to provide interactive ATM-based services," *Proc. IEEE GLOBECOM'95*, pp. 183-187, November 1995.

[94] P. Venkat Rangan, H. M. Vin, and S. Ramanathan, "Designing an On-Demand Multimedia Service," *IEEE Communications Magazine*, Vol. 30, no. 7, pp. 56-64, July 1992.

[95] N. Venkatasubramanian and S. Ramanathan, "Load Management in Distributed Video Servers," *International Conference on Distributed Computing Systems (ICDCS'97)*, pp. 528-535, May 1997.

[96] Y. Wang, J. C. L. Liu, D. H. C. Du and J. Hsieh, "Video File Allocation over Disk Arrays for Video-On-Demand," *Proc. ACM Multimedia'96*, pp. 160-163, 1996.

[97] W. D. Wei and C. L. Liu, "On a Periodic Maintenance Problem," *Operations Research Letters*, 2(2):90-93, 1983.

[98] W. Whitt, "The Pointwise Stationary Approximation for $M_t/M_t/s$ Queues is Asymptotically Correct as the Rates Increase," *Management Science*, Vol. 37, No. 3, pp. 307-314, March 1991.

[99] C.-S. Wu, G.-K. Ma, P.-N. Chen, "Architecture for Two-way Data Services Over Residential Area CATV Networks," *Proc. IEEE INFOCOM'97*, April 1997.

[100] Z.-X. Zhao, S. S. Panwar, D. Towsley, "Queueing Performance with Impatient Customers," *IEEE INFOCOM'91*, pp. 400-409, April 1991.

[101] R. Zimmermann, S. Ghandeharizadeh, "Continuous Display Using Heterogeneous Disk-Subsystems," *Proc. ACM Multimedia'97*, pp. 227-238, 1997.

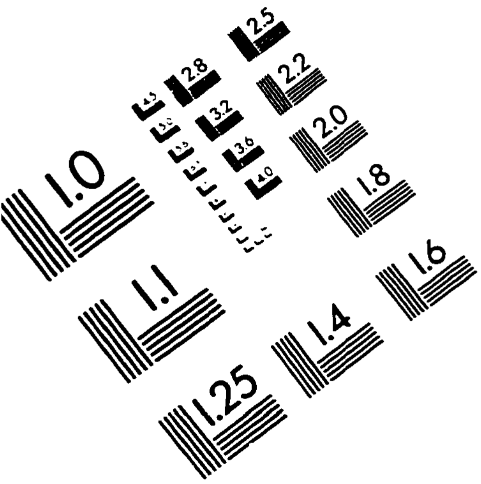
[102] G. Zipf, *Human Behaviour and the Principle of Least Effort*, Addison-Wesley, 1949.

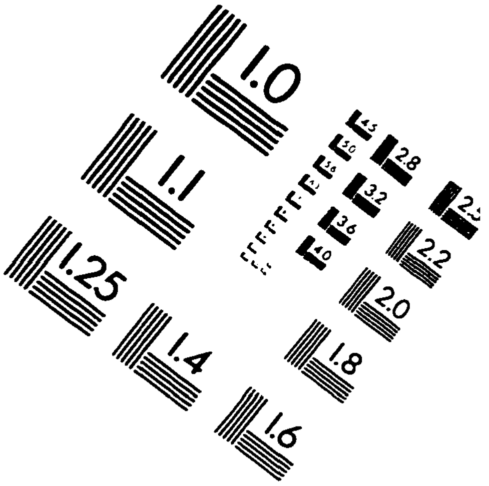
[103] *Seagate Product Overview*, October 1993.

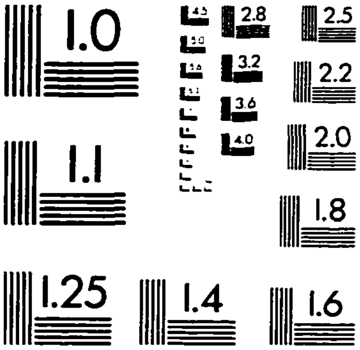
[104] "Digital Unveils Innovative Solutions for Ad Insertion and Near Video-On-Demand Applications," *Digital Press Release Homepage*, November 29, 1995. http://www.digital.com/info/PRHOME/.

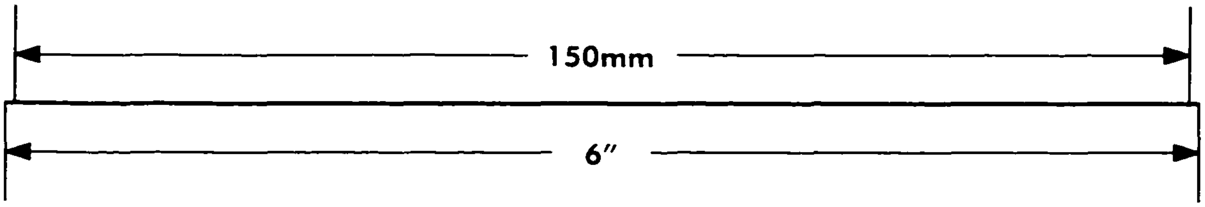
[105] "Sun MediaCenter Server: Marketing White Paper," *Sun MediaCenter Servers Homepage*, December 1996. http://www.sun.com/products-n-solutions/hw/servers/smc_external.html.

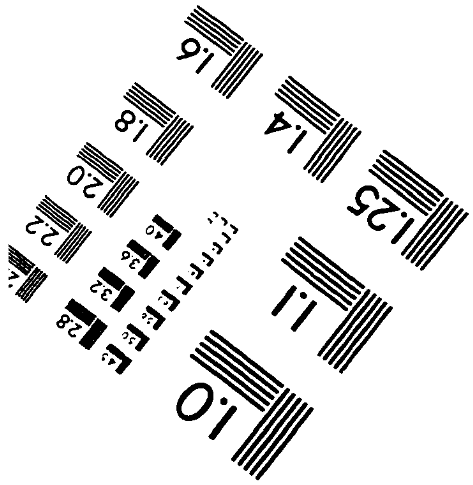
IMAGE EVALUATION
TEST TARGET (QA-3)

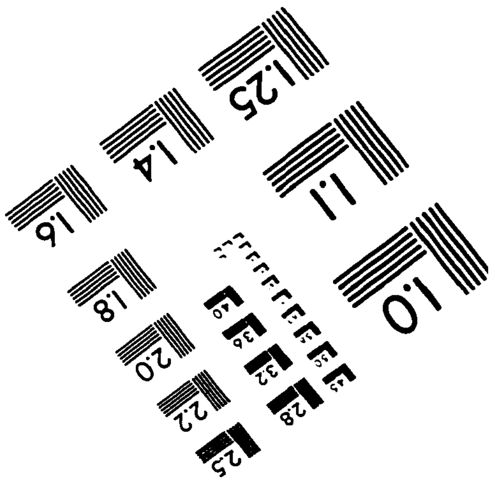












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989