Multicast Video-on-Demand Services *

Huadong Ma
College of Computer Science & Technology
Beijing University of Posts and Telecomm.
Beijing 100876, China
mhd@bupt.edu.cn

Kang G. Shin
Real-Time Computing Laboratory
Department of EECS
The University of Michigan
Ann Arbor, MI 48109-2122, USA
kgshin@eecs.umich.edu

ABSTRACT

The server's storage I/O and network I/O bandwidths are the main bottleneck of VoD service. Multicast offers an efficient means of distributing a video program to multiple clients, thus greatly improving the VoD performance. However, there are many problems to overcome before development of multicast VoD systems. This paper critically evaluates and discusses the recent progress in developing multicast VoD systems. We first present the concept and architecture of multicast VoD, and then introduce the techniques used in multicast VoD systems. We also analyze and evaluate problems related to multicast VoD service. Finally, we present open issues on multicast VoD as possible future research directions.

Keywords: Quality-of-Service (QoS), scheduling, VCR-like interactivity, multicast, Video-on-Demand (VoD)

1. INTRODUCTION

A typical Video-on-Demand (VoD) service allows remote users to play back any one of a large collection of videos at any time. Typically, these video files are stored in a set of central video servers, and distributed through high-speed communication networks to geographically-dispersed clients. Upon receiving a client's service request, a server delivers the video to the client as an isochronous video stream. Each video stream can be viewed as a concatenation of a storage-I/O "pipe" and a network pipe. Thus, sufficient storage-I/O bandwidth must be available for continuous transfer of data from the storage system to the network interface card (NIC), which must, in turn, have enough bandwidth to forward data to clients. Thus, a video server has to reserve sufficient I/O and network bandwidths before accepting a client's request. We define a *server channel* as the server resource required to deliver a video stream while guaranteeing a client's continuous playback.

This type of VoD service has a wide spectrum of applications, such as home entertainment, digital video library, movie-on-demand, distance learning, tele-shopping, news-on-demand, and medical information service. In general, the VoD service can be characterized as follows.

Long-lived session: a VoD system should support long-lived sessions; for example, a typical movie-on-demand service usually lasts 90–120 minutes.

High bandwidth requirements: for example, server storage I/O and network bandwidth requirements are 1.5 Mbps (3-10 Mbps) for a MPEG-1 (MPEG-2) stream.

Support for VCR-like interactivity: a client requires the VoD system to offer VCR-like interactivity, such as the ability to play, forward, reverse and pause. Other advanced interactive features include the ability to skip or select advertisements, investigate additional details behind a news event (by hypermedia link), save the program for a later reference, and browse, select and purchase goods.

QoS-sensitive service: the QoS that VoD consumers and service providers might care includes service latency, defection rate, interactivity, playback effects of videos, etc.

A conventional TVoD system uses one dedicated channel for each service request, offering the client the best TVoD service. However, such a system incurs very high costs, especially in terms of storage-I/O and network bandwidths. Moreover, such a VoD service has poor scalability and low performance/cost efficiency. Although the conventional approach simplifies the implementation, not sharing channels for client requests will quickly exhaust the network and the server I/O bandwidth. In fact, the network-I/O bottleneck has been observed in many earlier systems, such as Time Warner Cable's Full Service Network Project in Orlando [68], and Microsoft's Tiger Video Fileserver [12]. In order to support a large population of clients, we therefore need new solutions that efficiently utilize the server and network resources.

Clearly, the popularity or access pattern of video objects plays an important role in determining the effectiveness of a video delivery technique. Because different videos are requested at different rates and at different times, videos are usually divided into hot (popular) and cold (less popular),

*This work was partly done during Huadong Ma's visit to RTCL at the University of Michigan. The work was supported in part by the USA NSF under Grant EIA-9806280 and the Natural Science Foundation of China under Grant 69873006.

and requests for the top 10–20 videos are known to constitute 60–80% of the total demand. So, it is crucial to improve the service efficiency of hot videos.

Thus, requests by multiple clients for the same video arriving within a short time interval can be batched together and serviced using a single stream. This is referred to as *batching*. The multicast facility of modern communication networks [24, 25, 59] offers an efficient means of one-to-many[1] data transmission. The basic idea is to avoid transmitting the same packet more than once on each link of the network by having branch routers duplicate and then send the packet over multiple downstream branches. Multicast can significantly improve the VoD performance, because it

- reduces the required network bandwidth greatly, thereby decreasing the overall network load;
- alleviates the workload of the VoD server and improves the system throughput by batching requests;
- offers excellent scalability which, in turn, enables servicing a large number of clients; and
- provides excellent cost/performance benefits.

In spite of these advantages, multicast VoD (MVoD) introduces new and difficult challenges, as listed below, that may make the system more complex, and may even degrade a particular customer's QoS.

- It is difficult to support VCR-like interactivity with multicast VoD service while improving service efficiency.
- Batching makes the clients arriving at different times share a multicast stream, which may incur a long service latency (or waiting time) causing some clients to renege.
- A single VoD stream from one server cannot support clients' heterogeneity due mainly to diverse customer premise equipments (CPEs).
- A multicast session makes it difficult to manage the system protocol and diverse clients.
- Multicast VoD introduces the complex legal issue of copyright protection.

A multicast VoD system must therefore overcome the above drawbacks without losing its advantages. This paper critically reviews the recent progress in multicast VoD (including general VoD techniques) and discusses open issues in multicast VoD.

The remainder of this paper is organized as follows. Section 2 introduces the concepts and architectures of a multicast VoD system, and analyzes the problems in developing it. Section 3 reviews the implementations of multicast VoD. Section 4 discusses the issues related to multicast VoD service. Finally, Section 5 summarizes the paper and discusses open issues in implementing multicast VoD.

2. OVERVIEW OF MVOD SERVICE

[1]Multicast also covers multipoint-to-multipoint communication, but for our purpose in this paper, it suffices to consider only one-to-many communication.

Classification	Features
No-VoD	similar to broadcast TV, in which the user is a passive participant and has no control over the session.
PPV	in which the user signs up and pays for specific programming, similar to existing CATV PPV services.
QVoD	in which users are grouped based on a threshold of interest. Users can perform rudimentary temporal control activities by switching to a different group.
NVoD	functions like forward and reverse are simulated by transitions in discrete time intervals. This capability can be provided by multiple channels with the same programming skewed in time.
TVoD	the user has complete control over the session presentation. The user has full-function VCR capabilities, including forward and reverse play, freeze, and random positioning.

Table 1: Classification of VoD system

2.1 The taxonomy of VoD systems

A true VoD system supports a user to view any video, at any time and in any interactive mode. Based on the amount of interactivity and the ability of controlling videos, VoD systems are classified as Broadcast (No-VoD), Pay-Per-View (PPV), Quasi Video-On-Demand (QVoD), Near Video-On-Demand (NVoD), True Video-On-Demand (TVoD) [55] which are listed and compared in Table 1.

Obviously, TVoD is the most ideal service. For TVoD service, the simplest scheme of scheduling server channels is to dedicate a channel to each client, but it will require too many channels to be affordable. Since a client may be willing to pay more for TVoD service than for non-TVoD service, sharing a channel among clients is a reasonable way to improve the VoD performance and lower clients' cost. In fact, multicast can support all types of VoD services while consuming much less resources.

2.2 VCR interactivity of VoD

Interactivity is an essential feature of VoD service. After their admission, customers can have the following types of interactions: Play/Resume, Stop/Pause/Abort, Fast Forward/Rewind, Fast Search/Reverse Search, Slow Motion as identified in [54].

A TVoD service may also provide the support for other interactions such as *Reverse* and *Slow Reverse*, which correspond to a presentation in the reverse direction, at normal or slow speed. Usually, we don't consider them as part of the usual interactive behavior of a customer.

We classify interactive operations into two types: (1) *forward interactions*, such as Fast Forward and Fast Search; (2) *backward interactions*, such as Rewind, Reverse Search, Slow Motion, and Stop/Pause. This classification depends on whether the playback rate after interactive operations is faster than the normal playback or not. In order to understand the limited support provided by default in multicast VoD systems, one can identify two types of interactivity: *continuous* or *discontinuous* interaction [7]. Continuous in-

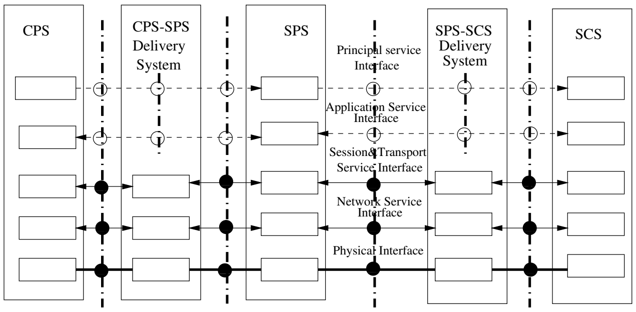

Figure 1: DAVIC reference model

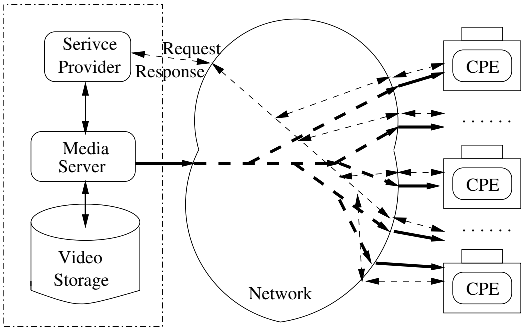

Figure 2: A multicast VoD system

teractive functions allow a customer to fully control the duration of all actions to support TVoD service, whereas discontinuous interactive functions allow actions to be specified only for durations that are integer multiples of predetermined time increment to support NVoD service. Note that the size of discontinuity is a measure of the QoS experienced by the customers from NVoD service.

From the implementation's perspective, we also categorize interactions as *interactions with picture* or *interaction without picture*. Fast/Reverse Search and Slow Motion are typical interactions with picture, whereas Fast Forward and Rewind are typical interactions without picture. In general, it is easier to implement interactions without picture because it requires less system resource.

2.3 The architecture of multicast VoD systems

2.3.1 The reference model of VoD systems

The Digital Audio-Visual Council (DAVIC) founded in 1994 is a non-profit organization which has charged itself with the task of promoting broadband digital services by the timely availability of internationally-agreed specifications of open interfaces and protocols that maximize interoperability across countries and applications or services. According to the DAVIC reference model shown in Figure 1 [27], a VoD system generally consists of the following entities:

Content Provider System (CPS) that owns and sells video programs to the service provider;

Service Provider System (SPS) is a collection of system functions that accept, process and present information for delivery to a service consumer system;

Service Consumer System (SCS) is responsible for the primary functions that allow a consumer to interact with the SPS and implemented in a customer premise equipment (CPE);

CPS-SPS and SPS-SCS network provider.

A consumer generates a request for service to the provider, who will obtain the necessary material from the program (content) provider and deliver it to the consumer using the network provider's facilities. The SPS acts as an agent for consumers and can access the various types of CPS. The network, CPS, and SPS can be the same organization, but they are generally different. DAVIC-based VoD systems have been developed, such as the one in [71], ARMIDATM [52], the NIST VoD system [45], KYDONIA [19], and the Broadband Interactive VoD system at Beijing Telecommunications.

The reference model is also suitable for specifying the architecture of MVoD systems. Consider a typical MVoD delivery system shown in Figure 2 [8, 35]. Consumers make program requests to the manager server (Service Provider). A request is received and queued by the manager server until the scheduler is ready to allocate a logical channel to deliver video streams from a video object storage to a group of consumers (multicast group) across a high-speed network. The manager server organizes the media server and network resources to deliver a video stream into a channel. A channel can be either a unicast or multicast channel. The media server receives consumer requests for video objects via the manager server, processes them, and determines when and which channels to deliver requested video objects to the consumers.

Each consumer accesses the system by a CPE which includes a set-top box (STB), a disk and a display monitor. A consumer is connected to the network via a STB, which selects one or more network channels to receive requested video objects according to the server's instructions. The received video objects are either sent to the display monitor for immediate playback, or temporarily stored on the disk which will later be retrieved and played back.

2.3.2 Hierarchical VoD systems

Large-scale VoD systems require the servers to be arranged as a distributed system in order to support a large number of concurrent streams. If the system is hierarchical, an end-node server handles the requests from a particular area, the next server in the hierarchy takes the requests over for end-node servers if they cannot handle them. This architecture provides the cost efficiency, reliability and scalability of servers. Generally, servers are either tree-shaped [60] or graph-structured [76, 77] in Figure 3. The graph-structured system often offer good QoS for handling the requests, but the managements of requests, videos and streams are complicated in the system. The tree-shaped system can easily manage requests, videos and streams, but it offers poorer QoS than the former. In order to evaluate the effectiveness of distribution strategies in such a hierarchy, the authors of [39] investigated how to reduce storage and network costs while taking the customers' behaviors into account.

Although some of hierarchical architectures are originally designed for unicast VoD services, they can also be used for multicast VoD to further improve the efficiency of service.

2.4 Problems with multicast VoD

Given below are the desired properties of a multicast VoD

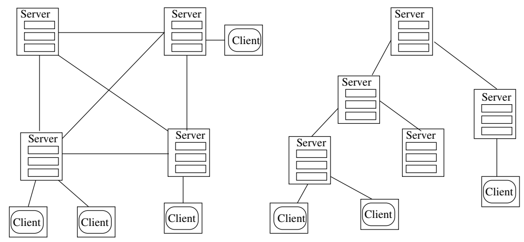

Figure 3: Hierarchical architecture of a VoD system

system.

Efficiency: The system should impose a minimal additional burden on the server and the network, and should sufficiently utilize critical resources on the server and the network.

Real-Time: The system should respond to the consumer requests and transmit the requested videos in real time.

Scalability: The system should scale well with the number of clients.

Interactivity: The system should provide the clients full control of the requested video by using VCR-like interactive functions.

Reliability: The system should be robust to failures in the server and the network, and easy to recover from failures. The transmission of messages and video streams should also be reliable.

Security: The system should provide efficient support for copyright protection in transmitting video streams to multiple clients.

Ability to deal with heterogeneity: The system should deal with heterogeneous networks and CPEs.

Fairness: The system should provide "fair" scheduling of videos with different popularities so as to treat all customers "fairly."

In order to meet the above requirements, we must solve the following key problems.

The first problem is how to deal with the coupling between system throughput and the batching interval. Increasing the batching interval can save server and network resources significantly at the expense of increasing the chance of customers' reneging: consumers are likely to renege if they are forced to wait too long, whereas shortening their waiting time will diminish the benefits of multicast VoD. In order to make this tradeoff, we must shorten all requests' waiting time while enabling each multicast session to serve as many consumers as possible.

The second problem is how to support scalability and interactivity. Support for full interactivity requires an "individualized" service for each customer by dedicating an interaction-(or I-) channel per consumer, which limits the scalability of multicast VoD. We need a fully-interactive on-demand service in multicast VoD systems without compromising system scalability and economic viability.

The third problem is how to guarantee customers' QoS with limited bandwidths. In multicast VoD, customers' QoS can be expressed in terms of the waiting time before receiving service (or *service latency*), the customers' *defection rate* due to long waits, and the VCR action blocking probability and playback effect. However, since system resources are limited, we must strive to maximize their utilization.

Moreover, the multicast VoD service generally favors popular videos, but how to serve the requests for unpopular videos in a multicast VoD framework is also of importance to the fairness of service.

3. IMPLEMENTATION OF MVOD

3.1 Storage organization

There are two types of servers: manager and media servers. The manager server (Service Provider) is responsible for billing and connection management, while the media server, the focus of this section, handles real-time retrieval and delivery of video streams.

The main challenge in the design of video server is how to utilize storage efficiently. When designing a cost-effective video storage, one must consider issues, such as placement of data on disks, disk bandwidth and disk-access QoS.

We consider the following main storage requirements.

- The VoD server requires a large storage capacity. A 100-minute MPEG-2 video with a transfer rate of 4 Mbps requires approximately 3 GBytes of storage space.
- Video objects are difficult to handle due to their large volume/size and stringent requirement of real-time continuous playback.

Most existing studies consider the use of multiple disks organized in the form of disk-farm or disk-array. A video server typically uses disk-array for large video data. When designing such a disk-array based VoD server, we must deal with several constraints on resource allocation to provide scalability, versatility, and load-balancing. Scalability is defined as the ability to absorb significant workload fluctuations and overloads without affecting admission latency, while versatility is defined as the ability to reconfigure the VoD server with a minimal disturbance to service availability. High-level versatility is also desirable for expandability, to ensure that new devices can be added easily. Each video can be stored on a single disk or stripped over multiple disks.

There are two basic types of storage organization. The first type completely partitions the storage among different movie titles. Such a storage system is said to have a *completely-partitioned* (CP) organization, and 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, which is said to have a *completely-shared* (CS) organization. VoD servers store movie titles using fine-grained striping (FGS) or coarse-grained striping (CGS) [65] of videos across disks in order to effectively utilize disk bandwidth. In FGS (similar to RAID-3), the stripe unit is relatively small and every retrieval involves all n disks that behave like a single logical disk with bandwidth nB (B is the bandwidth of one disk). In CGS, each retrieval block consists of a large stripe unit which is read from only a single disk, but different disks can simultaneously serve independent requests. CGS with parity information maintained on one or several dedicated disks corresponds to RAID-5 [11, 66].

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

Batching policy	Features	Comparison
Maximum Queue Length (MQLF) [21]	requests for the video with the largest number of pending requests to serve first.	maximizing the server throughput but unfairness to unpopular videos.
First-Come-First-Served -First (FCFS) [21]	the oldest request (with the longest waiting time) to serve next.	fairness but a lower system throughput.
Maximum Factored Queue Length First (MFQLF)[5]	the pending batch with the largest size weighted by the factor, (the associated access frequency)$^{-1/2}$,to serve next.	a throughput close to that of MQLF without compromising fairness.
Look-Ahead-Maximize -Batch (LAMB) [33]	a channel is allocated to a queue if and only if a head-of-the-line user is about to leave the system without being served	maximizing the number of admitted users in a certain time window but unfairness to some requests.
Group-Guaranteed Server Capacity (GGSC) [81]	server capacity is pre-assigned to groups of objects	meeting a given performance objective for the specific group.

Table 2: Multicast batching policies

of storage capacity. CS organizations ensure a very low latency and high storage utilization, but reconfigurations risk the availability of the entire VoD server. Studies in [3, 18, 32, 63] have shown that video striping improves disk utilization and load-balancing, and hence increases the number of concurrent streams. [3, 63] considered both CGS and FGS, and concluded that the former can support more concurrent video streams than the latter. This is because a disk has a relatively high latency for data access (10–20 ms), and a sufficient amount of video data must be transferred in each disk access in order to improve the utilization of the effective disk transfer bandwidth.

3.2 User-centered scheduling strategies

A conventional VoD system assumes the *user-centered* scheduling scheme [4, 83] in which a user eventually acquires some dedicated bandwidth. It can be achieved by providing (1) a sufficient bandwidth equal to an object consumption rate multiplied by the number of users, or (2) less bandwidth, for which the users compete by negotiating with a scheduler. The consumption rate of a video object is equal to the amount of bandwidth necessary to view it continuously. When a client makes a request to the server, the server sends the requested object to the client via a dedicated channel. This scheme incurs high system costs, especially in terms of server storage-I/O and network bandwidths. To maximally utilize these channels, researchers have proposed efficient scheduling techniques [20, 32, 43, 51, 61, 62, 64, 84]. These techniques are said to be "user-centered," because channels are allocated to users, not data or objects. These simplify the implementation, but dedicating a stream to each viewer will quickly exhaust the network-I/O bandwidth.

3.3 Data-centered scheduling strategies

To address the network-I/O bottleneck faced by the user-centered scheduling, one can use the data-centered scheduling which dedicates channels to video objects, instead of users. It allows users to share a server stream by batching their requests. That is, requests by multiple clients for the same video arriving within a short time interval can be batched together and served by using a single stream.

The data-centered scheme has the potential for dramatically reducing the network and server bandwidth requirements. The data-centered multicast VoD service can be either *client-initiated* or *server-initiated* [35]. In the client-initiated service, channels are allocated among the users and the service is initiated by clients, so it is also known as a *scheduled* or *client-pull* service. In the server-initiated service, the server channels are dedicated to individual video objects, so it is also called a *periodic broadcast* or *server-push* service. Popular videos are broadcast periodically in this scheme, and a new request dynamically joins, with a small delay, the stream that is being broadcast. In practice, it is efficient to use *hybrid batching* that combines the above two schemes.

3.3.1 Client-initiated multicast schemes

Using a client-initiated multicast, when a server channel becomes available, the server selects a batch to multicast according to the scheduling policies in Table 2.

The equally-spaced batching mechanism has a fixed maximum service latency and supports NVoD interactivity, but its usually-large service latency may cause some clients to renege. In order to reduce the service latency, *dynamic multicast* has been proposed, where the multicast tree is expanded dynamically to accommodate new requests.

For example, *Adaptive Piggybacking* [38] allows clients arriving at different times to share a data stream by altering the playback rates of in-progress requests (for the same object), for the purpose of merging their respective video streams into a single stream that can serve the entire group of merged requests. This approach can lower the service latency as compared to simple batching. But it is restrictive in that the variation of the playback rate must be within, say 5%, of the normal playback rate, or it will result in a perceivable deterioration of QoS. This limits the number of streams that can be merged.

Chaining [77] is also a generalized dynamic multicast technique to reduce the demand on the network-I/O bandwidth by caching data in the client's local storage to facilitate future multicasts. Thus, data are actually pipelined through the client stations residing at the nodes of the respective chaining tree, and the server serves a "chain" of client stations using only a single data stream. The advantage of chaining is that not every request has to receive its data directly from the sever. A large amount of video also becomes available from clients located throughout the network. This scheme scales well because each client station using the service also contributes its resources to the community. Hence, the larger the chaining trees, the more effective the application can utilize the aggregate bandwidth.

The authors of [16] present *stream tapping* that allows a client to greedily "tap" data from any stream on the VoD server containing video data s/he can use. This is accomplished through the use of a small buffer on the CPE and requires less than 20% of the disk bandwidth used by conventional systems for popular videos.

To eliminate the service latency, *patching* was introduced in [41]. The objective of patching is to substantially improve the number of requests each channel can serve per time unit, thereby sufficiently reducing the per-customer system cost. In the patching scheme, channels are often used to patch the missing portion of a service or deliver a patching stream, rather than multicasting the video in its entirety. Given that there is an existing multicast video, when to schedule another multicast for the same video is crucial. The time period after a multicast, during which patching must be used, is called the *patching window* [14]. Two simple approaches to setting the patching window are discussed in [41]. The first one uses the length of the video as the patching window. That is, no multicast is initiated as long as there is an in-progress multicast session for the video. This approach is called the *greedy patching* because it tries to exploit an in-progress multicast as much as possible. However, an overgreed can actually reduce data sharing [41]. The second approach, called the *grace patching*, uses a patching stream for the new client only if it has enough buffer space to absorb the skew. Hence, under grace patching, the patching window is determined by the client buffer size. Considering such factors as video length, client buffer size, and request rate, the authors of [15] generalized patching by determining the optimal patching window for each video. An improved form of patching, called as the *transition patching* [15], uses either a patching stream or a transition stream and improves performance without requiring any extra download bandwidth at the client site. Other optimal patching schemes were described in [30, 74]. In patching, a client might have to download data on both regular multicast and patching channels simultaneously. To implement patching, a client station needs three threads: two data loaders to download data from the two channels, and a video player to play back the video.

The *controlled CIWP* (Client-Initiated-With-Prefetching) [35] is another multicast technique similar to patching and tapping for near instantaneous VoD service. The novelty of the controlled CIWP is that it uses a threshold to control the frequency of multicasting a complete video stream. It uses simple FCFS channel scheduling so that a client can be informed immediately of when its request will begin service.

3.3.2 Server-initiated batching

In server-initiated batching, the bandwidth is dedicated to video objects rather than to users. Videos are decomposed into segments which are then broadcast periodically via dedicated channels, and hence, it is also called *periodic broadcast*. Although the worst-case service latency experienced by any subscriber is guaranteed to be less than the interval of broadcasting the leading segment and is independent of the current number of pending requests, this strategy is more efficient for popular videos than for unpopular ones due to the fixed cost of channels.

One of earlier periodic broadcast schemes was the *Equally-spaced interval Broadcasting* (EB) [21]. Since it broadcasts a given video at equally-spaced intervals, the service latency can only be improved linearly with the increase of the server bandwidth. The author of [10] also proposed the staggered VoD which broadcasts multiple copies of the same video at staggered times. To significantly reduce the service latency, *Pyramid Broadcasting* (PB) was introduced in [82]. In PB, each video file is partitioned into the segments of geometrically-increasing sizes, and the server capacity is evenly divided into K logical channels. The i-th channel is used to broadcast the i-th segments of all videos sequentially. Since the first segments are very small, they can be broadcast more frequently through the first channel. This ensures a smaller waiting time for every video. A drawback of this scheme is that a large buffer — which usually corresponds to more than 70% of the video — must be used at the receiving end, requiring disks for buffering. Furthermore, since a very high transmission rate is used for each video segment, an extremely high bandwidth is required to write data to the disk as quickly as it receives the video. To address these issues, the authors of [4] proposed a technique called *Permutation-based Pyramid Broadcasting* (PPB). PPB is similar to PB except that each channel multiplexes its own segments (instead of transmitting them sequentially), and a new stream is started once every short period. This strategy allows PPB to reduce both disk space and I/O bandwidth requirements at the receivers. However, the required disk size is still large due to the exponential nature of the data fragmentation scheme. The sizes of successive segments increase exponentially, thus causing the size of the last segment to be very large (typically more than 50% of the video). Since the buffer sizes are determined by the largest segment, using the same data fragmentation scheme proposed for PB limits the savings achievable by PPB. In PPB, a client needs to tune in different logical subchannels to collect its data for a given data fragment if the maximum savings in disk space is desirable.

To reduce the disk costs in the client side, the authors of [42] introduced *Skyscraper Broadcasting* (SB) which uses a new data fragmentation technique and proposes a different broadcasting strategy. In SB, K channels are assigned to each of the N most popular objects. Each of these K channels transports a specific segment of the video at the playback rate. The progression of relative segments size on the channel, $\{1,2,2,5,5,12,12,25,25,52,52,105,105,\ldots\}$, is bounded by the width parameter W, in order to limit the storage capacity required at the client end. SB allows for simple and efficient implementation, and can achieve a low service latency while using only 20% of the buffer space required by PPB. The authors of [34] provided a framework for broadcasting schemes, and designed a family of schemes for broadcasting popular videos, called the *Greedy Disk-conserving Broadcasting* (GDB). They systematically analyze the resource requirements, i.e., the number of server broadcast channels, the client storage space, and the client I/O bandwidth required by GDB. GDB exhibits a tradeoff between any two of the three resources, and outperforms SB in the sense of reducing resource requirements. The *Dynamic Skyscraper Broadcasting* (DSB) in [28] dynamically schedules the objects that are broadcast on the skyscraper channels to provide all clients with a precise time at which their requested objects will be broadcast, or an upper bound on that time if the delay is small and reaps the cost/performance benefits of the skyscraper broadcasting.

The above broadcasting schemes generally assume that the client I/O bandwidths are limited to download data from

Scheduling strategy	Features	Typical methods
Client-initiated (scheduled, client-pull)	the channels are allocated among the users, the multicast tree can be expanded dynamically to accommodate new requests so that the service latency is minimized (ideally zero).	Adaptive Piggybacking [38], Patching [41] Chaining [77], Tapping [16] and Controlled CIWP [35], etc.
Server-initiated (periodic broadcast, server-push)	the channels are dedicated to video objects, the videos are divided into segments which are then broadcast periodically via dedicated channels, the worst-case service latency experienced by any client is less than the interval of broadcasting the leading segment	Only two download channels: EB [21], PB [82], PPB [4], SB [42], GDB [34], DSB [28],etc. More than two download channels: Harmonic broadcasting [47],Staircase scheme [48] and FB [46, 80], etc.
Hybrid scheduling	the overall performance is improved by combining client-initiated and server-initiated strategies.	Controlled multicast [35], Catching and Selective catching [36], etc.

Table 3: The summary of existing data-centered approaches

only two channels. If the client can download data from more than two channels, there are methods available that can efficiently reduce the service latency with less broadcasting channels. For example, a broadcasting scheme based on the concept of harmonic series is proposed in [47, 49]; the scheme doesn't require the bandwidth assigned to a video equal to a multiple of a channel's bandwidth. For a movie of length of D minutes, if we want to reduce the viewer's waiting time to D/N minutes, we only need to allocate $H(N)$ video channels to broadcast the movie periodically, where $H(N)$ is the harmonic number of N, i.e., $H(N) = 1 + \frac{1}{2} + \ldots + \frac{1}{N}$. The *staircase* scheme in [48] can reduce the storage and disk transfer-rate requirement at the client end. However, both the staircase and harmonic schemes cannot serve bufferless users. In [46, 50], a scheme called *Fast Broadcasting* (FB) is proposed, which can further reduce the waiting time and the buffer requirement. Using FB, if a STB does not have any buffer, its user can still view a movie insofar as a longer waiting time is acceptable. The authors of [80] proposed two enhancements to FB, showing how to dynamically change the number of channels assigned to the video and seamlessly perform this transition, and presenting a greedy scheme to assign a set of channels to a set of videos such that the average viewers' waiting time is minimal.

3.3.3 *Hybrid multicast scheduling*

All practical scheduling policies are guided by three primary objectives: minimize the reneging probability, minimize average waiting time, and be fair. It was shown in [21, 22, 35, 36] that a hybrid of the above two techniques offered the best performance. For example, the *Catching* proposed in [36] is a combination of periodic broadcast and client-initiated prefix retrieval of popular videos. There are many hybrid schemes to improve the overall performance of multicast VoD. The selective catching in [36] further improves the overall performance by combining catching and controlled multicast to account for diverse user access patterns. Because most demands are on a few very popular movies, more channels are assigned to popular videos. However, it is necessary (and important) to support unpopular videos. We assume that scheduled multicasts are used to handle less popular videos, while the server-initiated scheme is used for popular videos. In this approach, a fraction of server channels are reserved and pre-allocated for periodically-broadcasting popular videos. The remaining channels are used to serve the rest of the videos using some scheduled multicasts. This hybrid of server-initiated and client-initiated schemes achieves better overall performance.

The existing data-centered approaches are summarized in Table 3.

3.4 Multicast routing and protocols

There has been extensive research into multicast routing algorithms and protocol [17, 72, 85]. Multicast can be implemented on both LANs and WANs. Nodes connected to a LAN often communicate via a broadcast network, while nodes connected to a WAN communicate via switched networks. In a broadcast LAN, transmission from any one node is received by all the other nodes on the network, so it is easy to implement multicast on a broadcast LAN. On the other hand, it is challenging — due mainly to the problem of scalability — to implement multicast on a switched network. Today's WANs are designed to mainly support unicast communication, but in future, as multicast applications become more popular and widespread, there will be a pressing need to provide efficient multicast support on WANs. In fact, the multicast backbone (MBone) of the Internet is an attempt toward this goal.

For multicast video transmissions, one of the key issues is QoS routing which selects routes with sufficient resources to provide the requested QoS. For instance, the multicast VoD service requires its data throughput to be guaranteed at or above a certain rate. The goal of QoS routing is twofold: (1) meet the QoS requirements for every admitted connection, and (2) achieve global efficiency in resource utilization. In most cases, the problems of QoS routing are proven to be NP-complete [86]. Routing strategies can be classified as source routing, distributed or hierarchical routing. Some heuristic QoS routing algorithms have been proposed (see [17, 85] for an excellent survey of existing multicast QoS routing schemes).

In an effort to provide QoS for video transmissions, a number of services have been defined in the Internet. A Resource Reservation Protocol (RSVP) has been developed to provide receiver-initiated fixed/shared resource reservation for unicast/multicast data flows [13] after finding a feasible path/tree to satisfy the QoS requirements. Furthermore, a protocol framework for supporting continuous media has been developed: RTP (Real-Time Protocol) [73] provides support for timing information, packet sequence numbers and option specification, without imposing any additional error control or sequencing mechanisms. Its companion control protocol, RTCP (Real-Time Control Protocol), can be used for gathering feedback from the receivers, again according to the application's need.

3.5 The client-end system

Customer premise equipments (CPEs) include set-top boxes (STBs), disks, and display monitors, where a disk or a RAM is used as a buffer. As an example, the disk space of 100 MB can cache about 10 minutes of MPEG-1 video. Such a disk space costs less than $10 today. The high cost of a VoD system is due mostly to the network costs. For instance, the cost of networking contributes more than 90% of the hardware cost of the Time Warner's Full Service Network project.

The client's STB, from software perspectives, generally contains a main control thread, video stream receiver threads and a video player thread. A client is connected to the network via a STB. The main control thread processes the client's service request by sending a message indicating his desired video to the server. It then forks the video stream receiver threads to select one or more network channels to receive and decompress video data according to the server's instructions. The received video data are either stored on the disk or sent to the display monitor for immediate playback. The display monitor can either retrieve stored data from the disk or receive data directly from a channel.

The CPE buffer plays important roles as follows.

- Supporting the VCR interactions of a customer [2, 7, 70]. The interaction protocols for multicast VoD are designed by using the CPE buffer.
- Providing instant access to the stored video program so as to minimize the service latency [36, 69]. Preloading and caching, based on the video stored in the CPE buffer, can reduce the service latency.
- Reducing the bandwidth required to transmit the stored video [36, 69]. Because some video data reside in the CPE buffer, the overall bandwidth requirement of transmitting videos is reduced.
- Eliminating the additional bandwidth required to guarantee jitter-free delivery of the compressed video stream.

These functions are discussed in detail in the following subsections.

3.6 Support for interactive functions

One of the important requirements is to offer VCR interactivity. In order to support customers' interactive behavior in multicast VoD service, there have been efficient techniques proposed by a combination of tuning and merging as well as using the CPE buffer and I-channels (see Table 4). The authors of [10] introduced *tuning* in staggered VoD which broadcasts multiple copies of the same video at staggered times. Intelligently tuning to different broadcast channels is used to perform a user interaction. However, not all interactions can be achieved by jumping to different streams. Moreover, even if the system can emulate some interactions, it cannot guarantee the exact effect the user wants. Other solutions to VCR interactivity are proposed in [7] and [23], especially for handling a pause/resume request. Support for continuous service of pause operations was simulated for a NVoD server, but merge operations from I-channels to batching- (or B-) channels were either ignored [23] or did not guarantee continuity in video playout [7]. [7] proposed the use of the CPE buffer to provide limited interactive functions. In order to implement the interactivity of multicast VoD services, more efficient schemes have been proposed. For example, the SAM protocol [53] offers an efficient way for TVoD interactions, and all those introduced in Section 2 are provided by allocating the I-channels as soon as a VCR action request is issued. When playout resumes, the VoD server attempts to merge the users back into a B-channel by using a dedicated *synch buffer* located at access nodes and partially shared by all the users. Should this attempt fail, a request for a new B-channel is then initiated.

The drawback of the SAM protocol requires an excessive number of I-channels, thus causing a high blocking rate of VCR interactions. The authors of [2] improved the SAM protocol by using the CPE buffer and active buffer management, and hence, more interactions can be supported without I-channel allocation. The BEP (Best-Effort Patching) scheme proposed in [56] presents an efficient approach to the implementation of continuous TVoD interactions. Compared to the other methods, BEP aims to offer zero-delay (or continuous) service for both request admission and VCR interaction, whereas the SAM protocol just supports continuous VCR interactions without considering service admission. Moreover, BEP uses a dynamic technique to merge interaction streams with a regular multicast stream. This technique significantly improves the efficiency of multicast TVoD for popular videos.

The authors of [70] proposed another scheme called the *Single-Rate Multicast Double-Rate Unicast* (SRMDRU) to minimize the system resources required for supporting full VCR functionality in a multicast VoD system. This scheme also supports TVoD service, so customers can be served as soon as their requests are received by the system. It forces customers in unicast streams (on the I-channel) to be served by multicast streams again after they resume from VCR operations. The idea is to double the transmission rate of the unicast stream so that the customer of normal playback can receive the frame synchronized with the transmitting frame of a multicast group.

4. ISSUES RELATED TO MVOD SERVICE

4.1 QoS of multicast VoD

The effectiveness of a video delivery technique must be evaluated in terms of both the server and network resources required for delivering a video object and the expected service latency experienced by the clients. Reducing the service latency is an important goal in designing effective scheduling strategies. The existing dynamic multicast and periodic broadcast schemes reviewed in Section 3.3 are shown to achieve good performance.

Besides the dynamic scheduling schemes, other schemes, such as *caching* and *preloading*, have been proposed to reduce the service latency. In [29, 75], proxy servers are used to cache the initial segments of popular videos to improve the service latency. Because nearly all broadcast protocols assume that the CPE buffer is large enough to store up to 40 or 50 % of each video (about 50 minutes of a typical movie), the *partial preloading* proposed in [69] uses this storage to preload anticipated customers' requests, say, the first 3 minutes of top 16 to 20 videos. It will provide instantaneous access to these videos and also reduce the bandwidth required

Level of interaction	Features	Typical methods
NVoD	the interactive functions are simulated by transitions in discrete time interval.	tuning [10]
Limited TVoD	the continuous interaction times are limited by the available resource.	that supported by CPE buffer [7]
TVoD	full control the durations of all continuous interactions.	SAM [53], Improved SAM [2] BEP [56], SRMDRU [70]

Table 4: The summary of interaction schemes for multicast VoD

Solution	Features	Comparison
Active transcoding	data streams are individually transformed according to the specification of each requesting receiver	an administrative burden, difficult to deploy proxies
Layered multicast	encodes source data as a series of layers, sends different layers for different multicast groups.	complex adaptation scheme in client-end

Table 5: The solutions of handling client heterogeneity

to broadcast them as well as the extra bandwidth required to guarantee jitter-free delivery of the compressed video signal. It differs from proxy caching in that the preloaded portions of each video will reside inside the CPE buffer rather than at a proxy server.

The customers' defection rate is closely related to the service latency, and is inversely proportional to the server throughput, or an average number of service requests granted per program. The shorter the service latency, the lower the defection rate becomes and the higher the server throughput is. Another important QoS parameter is the VCR action blocking probability. All the existing multicast TVoD protocols covered in Section 3.6 aim to reduce the blocking probability or discontinuity of VCR interactions.

4.2 Client heterogeneity

As the multicast network expands, there will be various types of end devices ranging from simple palm-top personal digital assistants (PDAs) to powerful desktop PCs or HDTV receivers of multicast VoD. Since there will be multiple VoD transmission rates or paths, the sender alone cannot meet the possibly conflicting demands of different receivers. Distributing a uniform representation of the video to all receivers could cause low-capacity regions of the network to suffer from congestion, and some receivers' QoS cannot be met even when there are sufficient network resources to provide better QoS for these receivers.

In the context of VoD, scalability also applies to the server's ability to support the data requirements of multiple terminal types. One way to solve this problem is proxy-based transcoding, where data streams are individually transformed according to the specification of each requesting receiver [31]. However, it typically imposes an administrative burden because it is not transparent to end users. Proxies are also difficult to deploy because a user behind a constrained network link might not have access to the optimal location of a proxy. There was a proposal to use active networks to solve this problem by offering a common platform for such services as part of the basic network service model [79], but there remain many issues to be addressed before such an infrastructure can be deployed. Furthermore, transcoding proxies must be highly reliable and scalable which can be very costly [31].

Another efficient solution to heterogeneity is the use of layered media formats. This scheme encodes source data as a series of layers, the lowest layer being called the *base layer* and higher layers being called the *enhancement layers*. Layered encoding can be effectively combined with multicast transmission by sending different layers for different multicast groups. Consequently, a receiver using only the basic multicast service (i.e., joining and leaving multicast groups) can individually tailor its service to match its capabilities, independently of other receivers. This basic framework was later refined in a protocol architecture called *Receiver-driven Layered Multicast* (RLM) [58]. In RLM, a receiver searches for the optimal number of layers by experimentally joining and leaving multicast groups much in the same way as a TCP source searches for the bottleneck transmission rate with the slow-start congestion avoidance algorithm [44]. The receiver adds layers until congestion occurs and backs off to an operating point below this bottleneck.

The two solutions to heterogeneity are summarized in Table 5.

4.3 Fairness of multicast VoD service

Fairness is one of the performance metrics in VoD service, meaning that every client request should be fairly treated regardless whether it is for a hot video or not. In [41], the unfairness of a multicast VoD system is expressed as a function of the defection rate, that is, $\sqrt{\frac{\sum_{i=1}^{N}(d_i - \bar{d})}{N-1}}$, where d_i denotes the defection rate for video i, $\bar{d}$ is the mean defection rate, and N is the number of videos. Alternatively, this property can also be measured by video service latencies.

The fairness is mainly related to scheduling and resource allocation. When selecting a scheduling strategy, we make it as fair as possible. The fairness of certain batching schemes are surveyed in Section 3.3. However, scheduling strategies like various periodic broadcasts, are only for popular videos, and the fairness depends on the scheduling scheme used for cold videos and the bandwidth allocation among hot and cold videos. Unfortunately, there are only a very few attempts to analyze the fairness of existing scheduling schemes [41]. How to assure the fairness of practical scheduling schemes, particularly for hybrid schemes, and make the optimal bandwidth resource allocation are open issues.

4.4 Customer behavior

Understanding customer behaviors is necessary to efficiently design a multicast VoD system and take different strategies to different videos at different times. Modeling customers' behaviors includes video selection distribution, variations of video popularity, and the user interaction model.

4.4.1 *Video selection distribution*

For short-term considerations, most researchers assume that the popularity of videos follows the Zipf distribution [87], that is, the probability of choosing the i-th video is $\frac{\frac{1}{i^{1-z}}}{\sum_{j=1}^{N} \frac{1}{j^{1-z}}}$, where N is the total number of videos in the system, and z is called the *skew factor*. Typically, researchers assume that the skew factor be set to 0.271 [5, 21]. This number is obtained from the analysis of a user access pattern from a video rental store [5]. That is, most of the demand (80%) is for a few (10 to 20) very popular videos.

4.4.2 *Time-variation of video popularity*

In a real VoD system, request arrivals are usually nonstationary. Variations in the request rate can be observed on a daily basis, between "prime time" (e.g., 7 p.m.–10 p.m.) and "off-hours" (e.g., early morning). On a larger time scale (e.g., one week or month), movie popularities may change due to new releases or loss of customers' interest in current titles. At the same time, the different types of customers (e.g., children and adult) have different prime times. In [1], a time distribution model is expressed as sinusoidal:

$$\lambda(t) = \lambda_0 + Asin(\gamma t)$$

$$\lambda_m(t) = p_m\lambda_0 + p_m Asin(\gamma t) = \lambda_m + A_m sin(\gamma t)$$

where λ_0 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 models of nonstationarity have been proposed in [8, 39] for the long-term popularity of movie. We call these time-dependent changes of movie popularity the life-cycle of the movie. The authors of [39] observed that the long-term behavior of a movie follows an exponential curve plus a random effect. [8] also assumed that variations in workload are exponential functions with different average inter-arrival times.

4.4.3 *Interaction model*

Some interaction models have been proposed in [2, 26, 54]. In [26], the behavior of each user is modeled by a two-state Markov process with states PB (playback) and FF/Rew. The times spent by the user in PB and FF/Rew modes are exponentially distributed. The two-state model in [54] assumes that the user activity is in either the normal or the interaction state. But these two models are too simple to represent the details of user interactions. To be realistic, a model should capture three specific parameters for their potentially significant impacts on the performance of the VoD server: (1) the frequency of requests for VCR actions; (2) the duration of VCR actions; (3) the bias of interaction behavior. Considering these parameters, the authors of [2] proposed a VCR interaction model. In this model, a set of states corresponding to the different VCR actions are designed durations and probabilities of transitioning to neighboring states. The initial state is Play, and the interaction system randomly transits to other interactive states or stays in the Play state according to the behavior distribution. The user resides at each interaction state for an exponentially-distributed period of time.

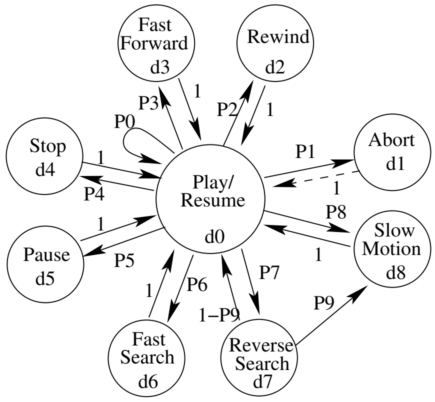

Figure 4: VCR interactive model

As shown in Figure 4, transition probabilities $P_i(i = 0, \ldots, 9)$ are assigned to a set of states corresponding to different VCR actions. It is important to notice that the above-mentioned parameters are captured by this representation of viewers' activity. Finding representative values for $P_i(i = 0, \ldots, 9)$ is still an open issue. For tractability, customers are classified to be *Very Interactive* (VI) or *Not Very Interactive* (NVI). Their interactions can be simulated by taking different parameter values [2].

4.5 Evaluation of multicast VoD systems

By batching multiple requests and serving them with a single multicast, the system capability for handling a large number of requests can be greatly improved at the expense of increased admission delay. For zero-delay admission and VCR interactions, more channels are required. It is important to evaluate multicast VoD in terms of throughput, resource requirement, and efficiency. These evaluation results will influence pricing, system management as well as resource sharing.

4.5.1 *The service throughput*

For NVoD service, we need to evaluate the throughput of multicast VoD. Common assumptions include the Poisson arrival of clients' requests, and customers' willingness to wait for a fixed amount of time, say, 5 minutes, before withdrawing their requests. The authors of [1] modeled the customers' patience with an exponential distribution, i.e., a customer agrees to wait for τ units of time or more with probability $p(\tau, \bar{\tau}) = e^{-\frac{\tau}{\bar{\tau}}}$, where $\bar{\tau}$ is the average time customers agreed to wait. In general, the patience rate $\alpha = \frac{1}{\bar{\tau}}$ can be assumed independently of the videos requested. Based on this assumption, the authors of [1] converted the problem of calculating the number of customers waiting between two consecutive services into that of making a transient analysis of the $M/M/\infty$ "self-service" queueing system with arrival rate λ, and self-service with a negative exponential distribution with rate α. They then derived the server throughput and the average loss rate for each movie. In [78], the user wait-tolerant behavior in some batching schemes, such as FCFS (first-come-first-serve), MQL (the maximum queue length), Max-Batch and Min-Idle, have also been investigated, the problem of maximizing the system throughput is formally discussed and the functional equation defining the optimal scheme is derived.

For TVoD service in multicast VoD, there is, to our best knowledge, no literature on evaluating the service throughput. It is still an open issue.

4.5.2 Bandwidth requirements of TVoD service

Some multicast VoD protocols can support TVoD services in zero-delay admission if enough channels are used. How to evaluate the channel requirements depends on the underlying multicast scheduling scheme. For example, [14] presents the optimal patching performance, and [15] addresses the bandwidth requirement of transition patching. [57] proposes a method for analyzing the user interactivity and evaluating the number of channels required for multicast TVoD service. The results determine the relationships among the clients' behaviors, the system resources, and the TVoD service protocol. However, rigorously analyzing the requirements of channels for TVoD protocols supporting zero-delay admission and full VCR interactions is still an open issue.

4.5.3 Bandwidth cost vs. QoS

Although most server-initiated multicast VoD schemes are not for providing TVoD service, they strive to make a better tradeoff between the channel cost and the service latency. For example, most periodic broadcasting schemes try to reduce the service latency and improve the throughput at low channel costs. The performance of related periodic broadcasting schemes have been discussed in [4, 34, 42, 83].

4.6 Other issues

Besides the mentioned issues, there also are other important issues in multicast VoD service, such as:

Copyright protection: A typical solution to video copyright protection is an encrypted point-to-point delivery to ensure that only paying customers receive the service, but it is inefficient for multicast VoD. In [40] a simple scheme is proposed to provide similar protection for the content, which can be used efficiently with multicast and caching. In that scheme, the major part of the video is intentionally corrupted and can be distributed via multicast connections, while the part necessary for reconstructing the original is delivered to each receiver individually. However, the efficient copyright protection for multicast VoD service is still an open issue.

Video replacement: The maintenance of storage is a main server management task in general VoD systems. Due to the limited storage capacity, it is necessary to replace old, unpopular videos by new popular videos. How to select videos to replace depends mainly on the hit ratio of videos. Popularity-based assignment and BSR(Bandwidth-to-Space Ratio)-based assignment are considered as the important policies. Moreover, the replacement operation for one video shouldn't affect the service of other videos residing in the same server. This problem is related to storage organization, and [3] discussed the effect of video partitioning strategies.

5. CONCLUSION AND FUTURE DIRECTIONS

As VoD service becomes popular and widely-deployed, consumers will likely suffer from network and server overloads. Network and server-I/O bandwidths have been recognized as a serious bottleneck in today's VoD systems. Multicast is a good remedy for this problem; it alleviates server bottlenecks and reduces network traffic, thereby improving system throughput and server response time. We discussed the state-of-art designs and solutions to the problems associated with multicast VoD. We also illustrated the benefits of multicast VoD, and feasible approaches to implementing it. Although multicast VoD can make significant performance improvements, there are still several open issues that warrant further research, including:

- Effective scheduling and routing schemes, and VCR-type interactions for multicast TVoD. Ad hoc schemes may achieve a better trade-off between QoS and system costs while preserving scalability and interactivity.
- Efficient active CPE buffer management. It is highly dependent on customers' behavior, and utilizing the CPE buffer will significantly improve QoS.
- Fairness of VoD service. There is a tradeoff between fairness and throughput for practical scheduling schemes, particularly for hybrid schemes. We need to achieve the optimal bandwidth resource allocation without loss of fairness.
- Knowledge of realistic customers' behavior is essential to the design of multicast VoD protocols and resource allocation. Other issues related to customers' behavior are throughput, scaling, and scheduling.
- An efficient theoretical framework for evaluating the performance of multicast VoD service, especially for modeling and analyzing multicast TVoD service to support both zero-latency admission and full VCR interactivity.
- Developing standard protocols for multicast VoD for practical applications. The existing protocols for multicast routing, scheduling and VCR interactions can be viewed as a basis to achieve this goal. The DAVIC protocol also provides a general reference framework, but protocols for multicast TVoD still need to be developed further.

6. REFERENCES

[1] E.L. Abram-Profeta and Kang G. Shin, "Scheduling video programs in near video-on-demand systems," *Proc. ACM Multimedia'97*, pp. 359–369, Seattle, Nov.1997.

[2] E.L. Abram-Profeta and Kang G. Shin, "Providing unrestricted VCR capability in multicast video-on-demand systems," *Proc. IEEE ICMCS'98*, June-July 1998.

[3] E.L. Abram-Profeta and Kang G. Shin, "A practical approach to resource allocation in video-on-demand servers," *Journal of Visual Communication and Image Representation*, 9(4):314-355,1998.

[4] C.C. Aggarwal, J.L. Wolf, and P.S. Yu, "A permutation-based pyramid broadcasting scheme for video-on-demand systems," *Proc. of IEEE ICMCS'96*, pp.118-126, Hiroshima, Japan, June 1996.

[5] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "On optimal batching policies for video-on-demand storage servers," *Proc. of IEEE ICMCS'96*, pp.253-258, Hiroshima, Japan, June 1996.

[6] C. C. Aggarwal, J. L. Wolf, and P.S. Yu, "On optimal piggyback merging policies for video-on-demand systems", *Proc. of ACM SIGMETRICS'96*, pp.200-209, Philadelphia, PA, May 1996.

[7] K.C. Almeroth and M.H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE JSAC*, 14(6), pp.1110-1122, Aug.1996.

[8] K.C. Almeroth *et al.*, "Long term resource allocation in video delivery system," *Proc. IEEE INFOCOM'97*, pp.1333-1340, Apr. 1997.

[9] ATM Forum, "Audiovisual multimedia services: video on demand specifications 1.0," *ATM Forum*, January 1996.

[10] R.O. Banker *et al.*, "Method of providing video-on-demand with VCR-like functions," *U.S. Patent 5357276*, 1994.

[11] E.W. Biersack and C. Bernhardt, "A fault tolerance video server using combined RAID-5 and mirroring," *Proc. MMCN'97, SPIE Vol.3020*, pp.106-117,1997.

[12] W. J. Bolosky *et al.*, "The tiger video fileserver," *Proc. of NOSSDAV'96*, Zushi, Japan, April 1996.

[13] R. Braden et al., "Resource reservation protocol (RSVP) -version 1 functional specification," *RFC 2205*, 1997.

[14] Ying Cai, K.A. Hua and K. Vu, "Optimizing patching performance," *Proc. of SPIE's Conference on Multimedia Computing and Networking (MMCN'99)*, pp.204-216, San Jose, January 1999.

[15] Ying Cai and K.A. Hua, "An efficient bandwidth-sharing technique for true video on demand systems," *Proc. ACM Multimedia'99*, pp.211-214, Orlando, Nov. 1999.

[16] S. W. Carter and D. D. E. Long, "Improving video-on-demand server efficiency through stream tapping," *Proc. IEEE ICCCN'97*, Las Vegas, pp.200-207, Sept. 1997.

[17] S. Chen and K. Nahrstedt, "An overview of quality of service routing for next-generation high-speed networks: Problems and solutions," *IEEE Network*, pp.64-79, Nov./Dec. 2000.

[18] A.L. Chervenak and D.A. Patterson, "Choosing the best storage system for video server," *Proc. of ACM Multimedia'95*, pp.109-119,1995.

[19] S. Christodoulakis *et al.*, "The KYDONIA multimeida information server," *Proc. the Second European Conference on Multimedia Services Application and Techniques (ECMSAT'97)*, Milan,1997.

[20] A. Dan, Y. Heights, and D. Sitaram, "Generalized interval caching policy for mixed interactive and long video workloads," In *Proc. of SPIE's Conf. on Multimedia Computing and Networking*, pp.344-351, San Jose, CA, January 1996.

[21] A. Dan, D. Sitaram, and P. Shahabuddin, "Scheduling policies for an on-demand video server with batching," *Proc. of ACM Multimedia'94*, pages 15-23, San Francisco, CA, October 1994.

[22] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic batching policies for an on-demand video server," *Multimedia Systems*, 4(3):112-121, June 1996.

[23] A. Dan *et al.*, "Channel allocation under batching and VCR control in video-on-demand systems," *Journal of Parallel and Distributed Computing*, 30:168-179, 1995.

[24] S. Deering, "Host extensions for IP multicasting," *Internet Engineering Task Force (IETF)*, RFC 1112, August 1989.

[25] S. Deering and D. Cheriton, "Multicast routing in datagram internetworks and extended LANs," *ACM Tran. on Computer Systems*, pp.85-111, May 1990.

[26] J.K. Dey-Sircar *et al.*, "Providing VCR capabilities in large-scale video servers," *Proc. of ACM Multimedia'94*, pages 25-32, San Francisco, CA, October 1994.

[27] Digital Audio-Visual Council, "DAVIC specifications 1.0 -1.4", *http://www.davic.org.*

[28] D.L. Eager and M.K. Vernon, "Dynamic skyscraper broadcasts for video-on-demand," *Proc. MIS'98*, Istanbul, Turkey, Sept. 1998.

[29] D.L. Eager, M. Ferris and M.K. Vernon, "Optimized regional caching for on-demand data delivery," *Proc. Multimedia Computing and Networking Conference (MMCN'99)*, San Jose, CA, Jan. 1999.

[30] D.L. Eager, M.K. Vernon, and J. Zahorjan, "Optimal and efficient merging schedules for video-on-demand servers," *Proc. ACM multimedia'99*, pp.199-202, Orlando, Nov. 1999.

[31] A. Fox *et al.*, "Adapting to network and client variation via on-demand, dynamic distillation," *Proc. ACM ASPLOS-VII*, pp.160-170, Oct. 1996.

[32] C. S. Freedman and D. J. Dewitt, "The SPIFFI scalable video-on-demand system," *Proc. of ACM SIGMOD'95*, pp.352-363, San Jose, CA, May 1995.

[33] Nelson L. S. da Fonseca and Roberto A. Facanha, "The Look-Ahead-Maximize-Batch Batching Policy, *Proc. of IEEE Globecom'99*, pages 354-358.

[34] L. Gao, J. Kurose, and D. Towsley, "Efficient schemes for broadcasting popular videos," *Proceedings of NOSSDAV'98*, Cambridge, UK, July 1998.

[35] L. Gao and D. Towsley, "Supplying instantaneous video-on-demand services using controlled multicast," *Proceedings of IEEE ICMCS'99*, pp.117-121, Florence, Italy, June 1999.

[36] L. Gao, Z.-L. Zhang and D. Towsley, "Catching and selective catching: efficient latency reduction techniques for delivering continuous multimedia streams," *Proc. ACM multimedia'99*, pp.203-206, Orlando, Nov. 1999.

[37] L. Golubchik, J. C. S. Lui, and R. Muntz, "Reducing I/O demand in video-on-demand storage servers", *Proc. ACM SIGMETRICS*, pp.25-36, Ottawa, Canada, May 1995.

[38] L. Golubchik, J. Lui, and Muntz, "Adaptive piggybacking: A novel technique for data sharing in video-on-demand storage servers," *Multimedia Systems*, 4(3):140-155,1996.

[39] C. Griwodz, M. Bar, and L. C. Wolf, "Long-term movie popularity models in video-on-demand systems or the life of an on-demand movie," *Proc. ACM Multimedia'97*, pp.349-357, Seatle, Nov. 1997.

[40] C. Griwodz *et al.*, "Protecting VoD the easier way," *Proc. ACM Multimedia'98*, pp.21-28, Bristol, U.K., Sept. 1998.

[41] K.A. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true video-on-demand services,"

Proc. ACM Multimedia'98, pp.191-200, Bristol, U.K., Sept. 1998.

[42] K.A. Hua and S. Sheu, "Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems," *Proc. ACM SIGCOMM'97*, pp.89-100, Sept. 1997.

[43] K.A. Hua, S. Sheu, and J. Z. Wang, "Earthworm: A network memory management technique for large-scale distributed multimedia applications," *Proc. of IEEE INFOCOM'97*,pp.990-997, Kobe, Japan, April 1997.

[44] V. Jacobson, "Congestion avoidance and control," *Proc. SIGCOMM'88*, pp.314-329, 1988.

[45] A. Jong, K. Hsing, D. Su, "A VoD application implemented in Java," *Multimedia Tools and Applications*, 5(2): 161-170, Sept. 1997.

[46] L.-S. Juhn and L.-M. Tseng, "Fast broadcasting for hot video access," *Proc. RTCSA'97*, pp.237-243, Oct.1997.

[47] L.-S. Juhn and L.-M. Tseng, "Harmonic broadcasting for video-on-demand service", *IEEE Transactions on Broadcasting*, 43(3):268-271, Sep.1997.

[48] L.-S. Juhn and L.-M. Tseng, "Staircase data broadcasting and receiving scheme for hot video service," *IEEE Transactions on Consumer Electronics*, 43(4):1110-1117, Nov.1997.

[49] L.-S. Juhn and L.-M. Tseng, "Enhanced harmonic data broadcasting and receiving scheme for popular video service," *IEEE Transactions on Consumer Electronics*, 44(2):343-346, May 1998.

[50] L.-S. Juhn and L.-M. Tseng, "Fast data broadcasting and receiving scheme for popular video service," *IEEE Transactions on Broadcasting*, 44(1):100-105, Mar.1998.

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

[52] S. D. Lago *et al.*, "ARMIDA TM: Multimedia applications across ATM-based networks accessed via internet navigation," *Multimedia Tools and Applications*, 5(2):133-146, Sept. 1997.

[53] W. Liao and V.O.K. Li, "The split and merge protocol for interactive video-on-demand," *IEEE Multimedia*, Oct.-Dec.1997, pp.51-62.

[54] V.O.K. Li *et al.*, "Performance model of interactive video-on-demand systems," *IEEE JSAC*, 14(6), pp.1099-1109, Aug. 1996.

[55] T.D.C. Little and D. Venkatesh, "Prospects for interactive video-on-demand," *IEEE Multimedia*, 1(3),1994, pp.14-24.

[56] Huadong Ma and Kang G. Shin, "A new scheduling scheme for multicast true VoD service," *Lecture Notes in Computer Science (Proc. PCM2001)*, Vol. 2195, pp.708-715, Springer, Oct. 2001.

[57] Huadong Ma and Kang G. Shin, "Performance analysis of the interactivity for multicast TVoD service," *Proc. IEEE International Conference on Computer Communications and Networks (IEEE ICCCN'01)*, Phoenix, Oct. 2001.

[58] S. McCanne and V. Jacobson, "Receiver-driven layered multicast,"*Proc. ACM SIGCOMM'96*, pp.117-130, Palo Alto, Aug. 1996.

[59] D. J. Marchok, C. Rohrs, and M. R. Schafer, "Multicasting in a growable packet (ATM) switch," *Proc. IEEE INFOCOM'91*, pp.850-858, 1991.

[60] J.P. Nussbaumer *et al.*, "Networking requirements for interactive video on demand," *IEEE JSAC*, 13(5), pp.779-787,1995.

[61] Y. Oyang *et al.*, "Design of multimedia storage systems for on-demand playback," *Proc. of Int'l Conf. on Data Engineering*, pp.457-465, Taipei, March 1995.

[62] B. Ozden *et al.*, "A low-cost storage server for movie on demand databases," *Proc. of the 20th Int'l Conf. on VLDB*, pp. 594-605, Santiago, Chile, Sept. 1994.

[63] B. Ozden, R. Rastogi, A. Silberschatz, "Disk striping in video servers environments," *Data Enginerring*, 18(4):4-16, 1995.

[64] B. Ozden *et al.*,"Demand paging for video-on-demand servers," *Proc. of IEEE ICMCS'95*, pp.264-272, Washington DC, May 1995.

[65] B. Ozden, R. Rastogi, and A. Silberschatz, "Disk striping in video server environment," *Proc. of IEEE ICMCS'96*, pp.580-589, 1996.

[66] B. Ozden, R. Rastogi, and A. Silberschatz, "Fault-tolerant architectures for continuous media servers," *Proc. of SIGMOD'96*, pp.79-90, 1996.

[67] J.C. Pasquale, G.C. Polyzos and G. Xylomenos, "The multimedia multicasting problem,"*Multimedia Systems*, 6:43-59, 1998.

[68] T.S. Perry, "The trials and travails of interactive TV," *IEEE Spectrum*, 33(4), pp.22-28, April 1996.

[69] J.-F. Paris, D. Long and P. E. Mantey, "Zero-delay broadcasting protocols for video-on-demand," *Proc. ACM Multimedia'99*, pp.189-197, Orlando, Nov. 1999.

[70] W.W.F Poon and K.T. Lo, "Design of multicast delivery for providing VCR functionality in interactive video-on-demand systems,"*IEEE Trans. on Broadcasting*, 45(1): 141-148, MAR 1999.

[71] S.M. Poon, B.S. Lee and C.K. Yeo, "A Davic-based video-on-demand system over ATM networks", *IEEE Transactions on Consumer Electronics*, 45(2), pp.345-355, May 1999.

[72] L. H. Sahasrabuddhe and B. Mukherjee, "Multicast routing algorithms and protocols: A tutorial," *IEEE Network*, pp.90-102, Jan./Feb. 2000.

[73] H. Schulzrinne *et al.*,"RTP: A transport protocol for real-time applications,"*Internet Request For Comments, RFC 1889*.

[74] S. Sen *et al.*, "Optimal patching schemes for efficient multimedia streaming," *Proc. IEEE NOSSDAV'99*, Basking Ridge, NJ, June 1999.

[75] S. Sen, J. Rexford, and D. Towsley,"Proxy prefix caching for multimedia streams," *Proc. IEEE INFOCOM'99*, New York, March 1999.

[76] C. Shahabi, M. Alshayeji, and S. Wang, "A redundant hierarchical structure for a distributed continuous media server," *Proc. IDMS'97 (LNCS 1309)*, Darmstadt, Germany, Sept.1997.

[77] S. Sheu, K.A. Hua and W. Tavanapong, "Chaining: a generalized batching technique for video-on-demand systems," *Proc. IEEE ICMCS'97*, pp.110-117, Ottawa,1997.

[78] Hadas Shachnai and P. S. Yu, "Exploring wait tolerance in effective batching," *Multimedia Systems*,

Vol. 6, pp.382-394, 1998.

[79] D.L. Tennenhouse *et al.*,"A survey of active network research,"*IEEE Communications Magazine*, 35, pp.80-86, Jan. 1997.

[80] Y.-C. Tseng *et al.*, "Data broadcasting and seamless channel transition for highly-demanded videos," *Proc. IEEE INFOCOM'2000*, pp.727-736, Tel-Aviv, March 2000.

[81] A. K. Tsiolis and M. K. Vernon, "Group-Guaranteed Channel Capacity in Multimedia Storage Servers", *Proc. of ACM SIGMETRICS'97*, pp.285-297.

[82] S. Viswanathan and T. Imielinski, "Pyramid broadcasting for video on demand service," *Proc. the SPIE Multimedia Computing and Networking Conference*, 2417, pp.66-77, San Jose, CA, 1995.

[83] S. Viswanathan and T. Imielinski, "Metropolitan area video-on-demand service using pyramid broadcasting," *Multimedia Systems*, 4(4):197-208, August 1996.

[84] H. M. Vin and P. V. Rangan, "Designing a multiuser HDTV storage server," *IEEE JSAC*, 11(1):152-164, Jan. 1993.

[85] B. Wang and J. C. Hou, "Multicast routing and its QoS routing: problems, algorithms, and protocols," *IEEE Network*, pp.22-36, Jan./Feb. 2000.

[86] Z. Wang and J. Crowcroft, "QoS routing for supporting resource reservation," *IEEE JSAC*, 14,pp.1228-1234, Sept. 1996.

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