Providing Unrestricted VCR Functions in Multicast Video-on-Demand Servers [*]

Emmanuel L. Abram-Profeta and Kang G. Shin
Real-Time Computing Laboratory
EECS Department, University of Michigan
Ann Arbor, Michigan 48109–2122
{*abram,kgshin*}*@eecs.umich.edu*

Abstract

Full support for interactive VCR functions can be achieved only by dedicating a channel for each customer. Unfortunately, channel dedication should not be used in a multicast video-on-demand (VoD) system, because it degrades scalability, and hence economical viability, which is usually achieved by multicasting videos. We, therefore, propose and evaluate a fully-interactive, yet scalable, on-demand service in multicast VoD systems. In the proposed framework, support for interactive operations is provided by (1) allocating a fraction of the VoD-server's channel capacity to handle interactive operations which would otherwise be blocked, and (2) using partial caching of programs in customers' premise equipment (CPE) to merge customers back in synchronization with broadcast channels. Our evaluation results indicate that batching policies under the proposed framework can make an attractive tradeoff between scalability and customers' QoS.

1. Introduction

The need to batch requests for the same movie title together has long been recognized for scalability, commercial success and immediate deployment of a multicast video-on-demand (VoD) system [4, 5]. As mentioned in [1, 6], some multicast VoD systems already provide customers with a framework for limited and scalable VCR capability. In "Near-VoD" (NVoD) or "Quasi-VoD" (QVoD) systems [1], for instance, video programs for the same movie title are sourced with (quasi) regularly-staggered intervals, called *phase offsets*, and 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 caching a small amount of video data (e.g., 5 minutes' worth of video) in a buffer located close to the client, such as in the customers' premise equipment (CPE). This buffer can then 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. In such VoD systems, however, 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 paper, we address how to provide a fully-interactive on-demand service in multicast VoD systems without compromising system scalability and economical viability.

In this paper, we (1) propose mechanisms for unrestricted VCR functionality with minimal degradation of system scalability, and (2) *integrate* 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 dedication of "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. We show that the CPE buffer can be actively employed to merge a customer served by an I-channel back in synchronization with a "batching" channel (B-channel), by simply prefetching frames or groups of frames while the I-channel is serving the customer in playback mode [11]. 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, or 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 sev-

---

[*] The work reported in this paper was supported in part by the National Science Foundation under Grant MIP-9203895 and the Office of Naval Research under Grant N00014-94-1-0229.

eral independent factors, such as: (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. This paper aims at providing a better understanding and guidance in the way these factors interact.

The paper is organized as follows. As a starting point, we first outline 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 experimental setup used to evaluate VCR functionality. For the sake of realism, we introduce an idealized model of customers' interaction behavior, which captures several key features, such as frequency and durations of VCR actions, and bias towards "forward" or "backward" interactions. Finally, Section 4 describes preliminary simulation results on a carefully-chosen batching policy under various conditions of customers' behavior and resource allocation. The paper concludes with Section 5.

2. Support for Limited VCR Functionality

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

2.1. Definition of Interactive Behavior

Customers wait in line until the VoD server decides when to start program transmission based on such constraints 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 following types of interactions, originally identified in [10]:

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). Note that 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.

In order to understand the limited support provided by default in multicast VoD systems, one can discriminate two levels of interactivity according to the linearity in playback experienced by customers [6]. *Continuous* interactive operations allow a customer to fully control the duration of interaction, whereas *discontinuous* interactive operations such as fast forward and rewind, can only be specified for durations that are integer multiples of a predetermined time increment. By caching a limited amount of video data, the CPE buffer partially supports both continuous and discontinuous actions, and can be accessed with very low latency during interaction [6]. Multicast groups are then 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.

Discontinuities in VCR actions happen in two possible scenarios. First, customers may suddenly exceed the CPE-buffer capacity while performing a continuous action, e.g., pausing for too long. In that case, the VoD server may transfer the customer to another multicast group. Second, in fast forward and rewind, a customer may specify the length of video s/he wants to skip, in which case the VoD server will determine the multicast group 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.

2.2. Continuous Service of VCR Actions

The CPE buffer is sized depending on its affordability. For example, 5 minutes' worth of MPEG-1 compressed video at 1.5 Mbps represents approximately 56.25 MB, which, stored in DRAM for fast access, should cost around $\$200$ in 1997. Thus, a small buffer (e.g., 1 minute) is likely to be affordable 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 Fig. 1. The CPE buffer can be seen as a *sliding window* over the largest *usable* sequential portion of the video (e.g., 7000 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 linear access in the portion of video program currently held by the CPE buffer — and the oldest frame. (The most recent frame usually corresponds to the

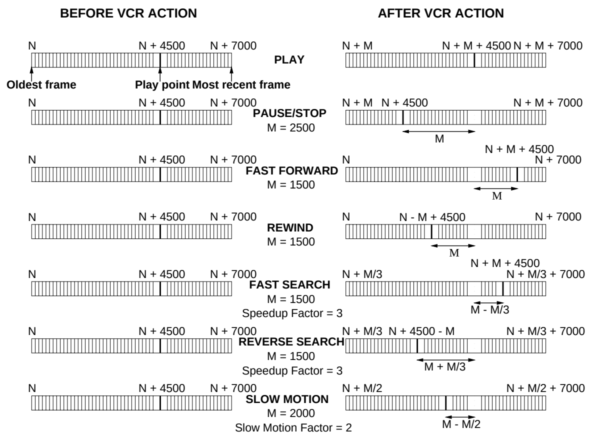

Figure 1. Displacement of the play point within the CPE buffer

most recently prefetched frame from the multicast group.) 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 (or "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 Fig. 1 — and throughout the paper — 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 an arbitrary 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 in physical storage. We deliberately ignore 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 Fig. 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 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 play point's position 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 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 as backward interactions.

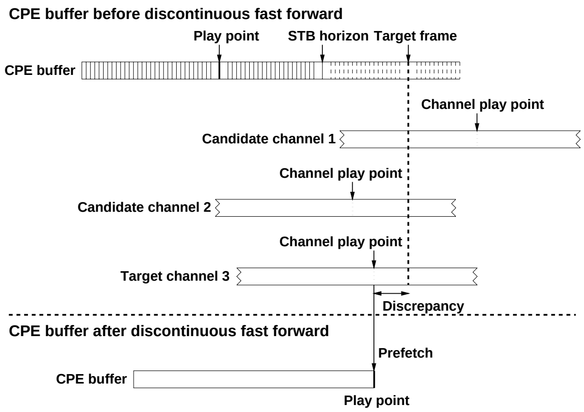

Figure 2. Discontinuous fast forward action.

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 Fig. 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 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

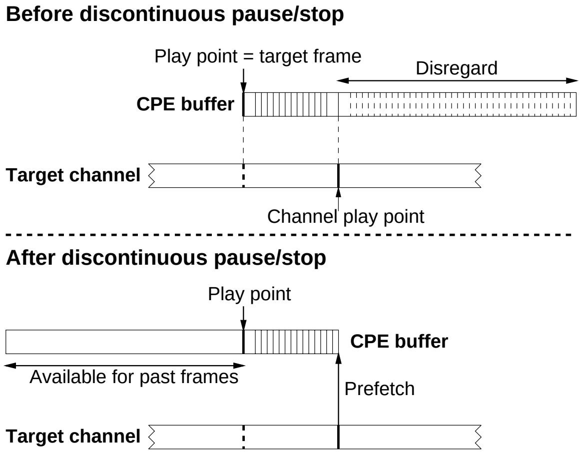

Figure 3. Discontinuous pause/stop: case 1.

determine the channel whose play point is the closest to the target frame requested by the customer. This operation is illustrated in Fig. 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 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 cause discontinuity in the playback seen by the viewer. In the situation represented in Fig. 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 Fig. 4 (cases 2 and 3), the target channel is chosen to minimize the jump experienced when the customer resumes. The CPE buffer content has to be entirely discarded. 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.

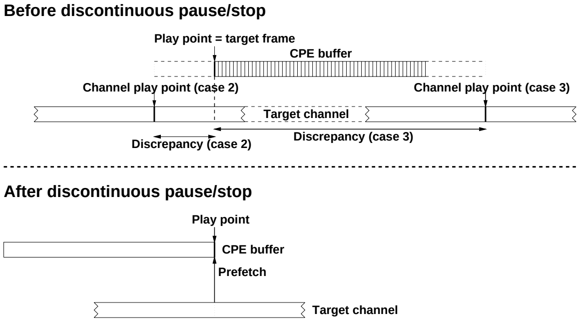

Figure 4. Discontinuous pause/stop: cases 2 and 3.

The number of channels allocated to a particular movie title will determine the discontinuity experienced by customers of that movie in discontinuous VCR actions. If very few channels (e.g., $2-3$) 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 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.

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. In this section we propose a framework for full-fledged, yet scalable support for interactive operations by (1) allocating I-channels on-demand to handle interactive operations which would otherwise be blocked or served in a discontinuous manner, and (2) using the CPE buffer to merge the customer back in synchronization with a B-channel once playback is resumed within an I-channel.

3.1. Related Work

VoD systems combining requests batching and VCR functionality were initially presented in [6], [8], and [11]. In [6] and [8], support for continuous service of pause operations was simulated for an NVoD server but merge operations from I- to B-channels were either ignored ([8]) or did not guarantee continuity in video playout ([6]). In the *split-and-merge (SAM) protocol* [11], all interaction types introduced in Section 2.1 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 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. Although the SAM protocol, and to a lesser extent the schemes presented in [6] and [8], dramatically increase the capacity of the system in comparison with true 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 different 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* [5]. 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 batching policies lack scalability [5, 2] and 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 [8] 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 [8] that given arbitrary 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.

Next, the partially-shared synch buffer, statically allocated by the VoD server and located at the access nodes, is used exclusively for merge operations. Had the synch buffer capacity been used similarly to a CPE buffer as in Section 2.2, more interactions would have been served without I-channel allocation. 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 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, poor buffer utilization in SAM leads to a blocking probability and delay in VCR actions that grow *linearly* in traffic intensity. Unlike the shared-buffer approach presented in [11], we will show in Section 3.2 that active use of a CPE buffer provided by the customers themselves is needed to achieve a higher QoS and scalability in VCR actions without incurring more buffer space than in the SAM protocol. Although a shared buffer may seem a cheaper alternative, decentralized management of non-shared buffers is better suited to large-scale deployment since (1) no extra buffer, bandwidth, and processing power are needed as the service provider expands, (2) buffer costs are divided among customers.

Finally, 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 2. In the SAM protocol, blocked interaction requests are queued, and, therefore, experience unpredictable delays which limit the "on-demand" nature of the service.

3.2. Framework for Scalable Interactions

3.3. 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 Fig. 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 will be blocked, whereas in other cases, the VCR action will be served in a discontinuous fashion as explained in Section 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). Several techniques have been proposed to implement fast and reverse search operations at

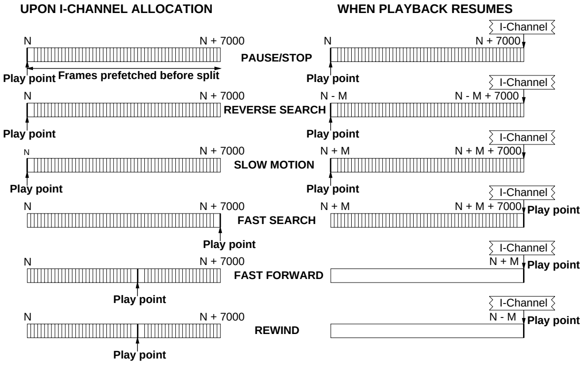

Figure 5. CPE buffer before I-channel allocation and after playout resumes.

n times the normal playback rate [7, 9, 13]. They usually differ in the additional resources required (e.g., file format, storage, network support), in the flexibility in the choice of a browsing speed, and in the discontinuity felt by the user. While it is beyond the scope of this paper to determine which scheme is better suited for VoD service, 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 or if fast access is performed by retrieving a version of the movie intended for that purpose [9, 13]. 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. For a simple description of CPE-buffer management, we also assume that all frames are of the same size. However, most of the mechanisms can be easily adapted to VBR video or to CBR transport of video with different frame sizes.

The right side of Fig. 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 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 $SP \times R$ without changing the relative position of the play point and the number of future frames within the CPE buffer. In the case of 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.

3.4. 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 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 Fig. 5 — we can distinguish two situations. First, as illustrated in Fig. 6, 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 Fig. 6[1]. 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

---

[1] Note that the merge operation in this case is similar to a discontinuous pause/stop, case 1 in Section 2.3

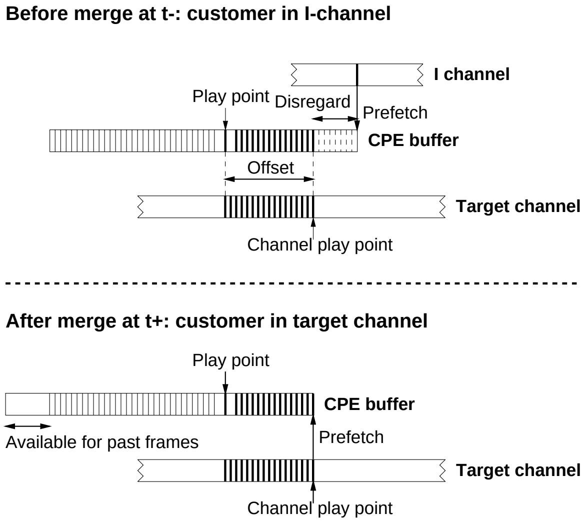

Figure 6. Instantaneous merge operation.

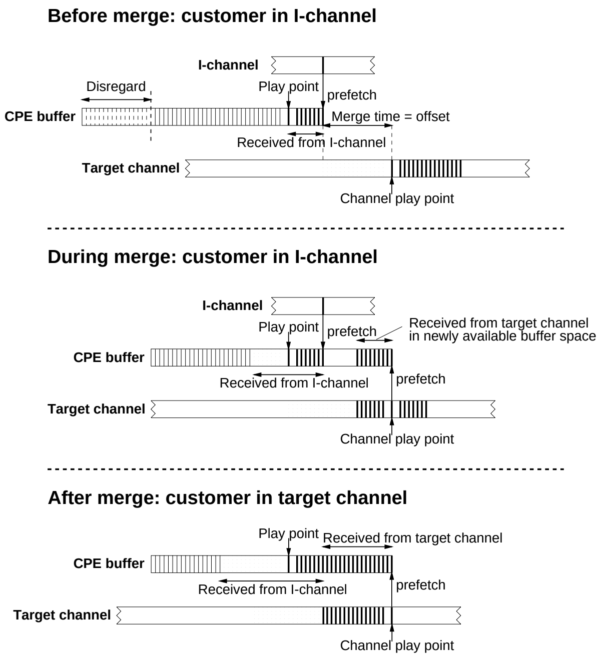

Figure 7. General merge operation.

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 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 us 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 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 Fig. 7. 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 an eligible 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 Fig. 1 when the customer is in playback mode, and as indicated in Section 3.3 during VCR actions.

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 re-

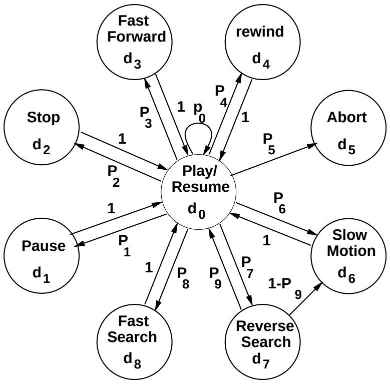

Figure 8. Transition diagram for the customer activity model.

verted 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.

4. Numerical Results

4.1. The Simulation Environment

Customers' behavior can be divided into *admission* and *interactive* behavior. We first assume that the aggregated arrival of customers' requests is a Poisson process with parameter λ. Given a concurrent channel capacity s, the traffic intensity is $\rho = \frac{\lambda L}{s}$, where L is the average length of a movie. We assume in the rest of this paper that $N = 10$ different movie titles of duration $L = 100$ minutes are made available by the VoD server and we adopt Zipf's law [4] as the movie-popularity model.

Once admitted, customers interact with the VoD server according to the customer activity model depicted in Fig. 8. In this model, 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 presented in Section 2.1. 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). Note that the set of transitions depicted in Fig. 8 is an arbitrary example used to provide intuition about the key aspects of the problem. Other sets of states and transitions are perfectly possible, if they prove to be closer to real behavior. Unlike the models of customers' interaction behavior that have been proposed [10], our model captures, for the sake of realism, three specific parameters for their potentially significant impact on the performance of the VoD server: (1) the level of interactivity, or frequency of requests for VCR actions; (2) the duration of VCR actions; (3) the *bias* of interaction behavior, which indicates whether interactive operations are dominated by backward or forward actions, or evenly distributed between the two. 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 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.

$Action$	P/R	A	S	RW	FF	FS
P/R	0.75	0.02	0.035	0.04	0.04	0.04

Table 1. Transition probabilities from play/resume.

Finding *representative* values for Fig. 8 is still an open issue, since, to the best of our knowledge, there are no published realistic (or empirically verified as such) models of interactive customers. Such usage statistics could be collected in visualization laboratories, or libraries with traditional video equipment, by monitoring the amount of time each user spends performing a particular operation, counting transitions from one state to another, and computing their relative frequencies. With the exception of the duration in play/resume state being fixed at 10 minutes and in abort (A) state (of null duration), VCR actions are assumed to be of equal length 1 minute. As represented in Fig. 8, the transition probability from any other state than play/resume (P/R) 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., $P_6 = 0$). We also assume that that the bias in VCR actions is *neutral*. Transition probabilities are summarized in Table 1. (Values for reverse search and pause are not represented as they are assumed identical to rewind (RW) and stop (S), respectively.) 11 other interaction behaviors are defined in [3], depending on VCR actions durations and bias, and customers' interactivity level.

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; and (4) served in a discontinuous fashion[2]. This classification allows us to evaluate both customers' QoS in service received,

---

[2] 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).

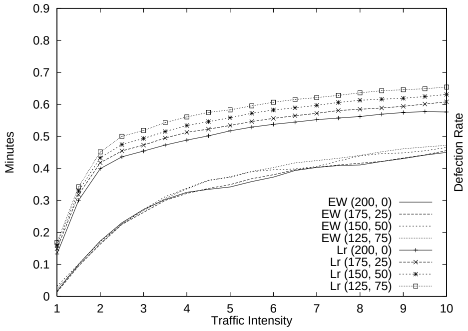

Figure 9. Admission latency and defections.

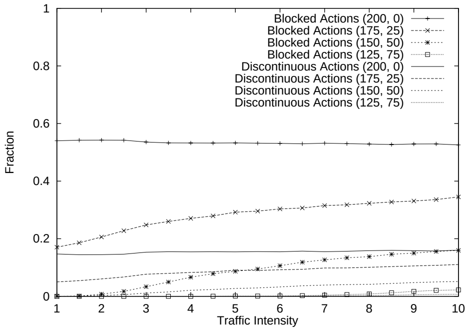

Figure 10. Blocked and discont. actions.

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.

4.2. Choice of a Partition between B- and I-Channels

A good batching policy should protect customers' QoS and server throughput from unpredictable or sustained variations in traffic intensity. The ability to handle an arbitrarily large number of customers without loss of performance is known as "scalability." Due to length constraints, we shall restrict our illustration of the proposed framework to *"QVoD-enhanced" NVoD*, scalable batching policy introduced in [1, 2] as an alternative to both on-demand and NVoD policies. In "QVoD-enhanced" NVoD, the B-channel capacity s is first partitioned among N movie titles into $[s_1, \cdots, s_N]$ as in NVoD, by optimizing an arbitrary predetermined objective as shown in [1] (e.g., minimize the average phase offset, the defection rate for partially-patient customers, or make a tradeoff between throughput and average phase offset). Then, a vector of optimal thresholds $\overline{K}^{opt} = [K_1^{opt}, \cdots, K_N^{opt}]$ is defined so that the s_i channels dedicated to a particular movie title i are initiated only as soon as a certain threshold K_i^{opt} on the number of pending requests has been reached. The choice of optimal thresholds usually depends on the system operating point (e.g., traffic intensity, customers' patience) and can be set so that either customer's service latency, or defection rate are minimized. We chose "QVoD-enhanced" NVoD as it usually exhibits a superior performance over other scalable batching policies in terms of customers' admission QoS, for identical breakdown of VCR actions. Although performance is very sensitive to threshold tuning, most conclusions presented hereafter hold for other scalable policies such as NVoD.

Our methodology compares performance in batching, discontinuous operations, and continuous operations. If more B-channels are allocated, a lower defection rate and possibly 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 for a capacity of 200 channels in four different partitions, $(B, I) = (200, 0)$, $(175, 25)$, $(150, 50)$ and $(125, 75)$. Figs. 9, 10, and 11 represent both admission and service performance. We assumed partially-patient customers who willing to wait for an exponentially-distributed period of time of mean 1 minute, and accordingly, a CPE-buffer size set to 1 minute. As can be seen from Fig. 10, partitions $(150, 50)$ and $(125, 75)$ dramatically reduce the fractions 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 (L_r), (2) no change in the average latency (EW), and (3) no loss in scalability, as indicated by the quasi-parallelism that can be observed, for the same batching policy, among curves corresponding to different partitions. Clearly, and as confirmed in other experimental scenarios presented in [3], a multicast VoD server can provide unrestricted support for interactive operations with minor impact on system scalability.

4.3. Comparison with the SAM Protocol

Fig. 12 compares 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 Fig. 7, and for stop and pause actions. (Note that our experiment slightly improves the SAM protocol in comparison with the scheme originally proposed in [11], by considering scalable batching, and by isolating respective allocations of B- and I-channels. The blocking probability documented in [11] is linear in traffic intensity.) The fractions of blocked and discontinuous actions are dramatically increased in the SAM case (up to 33% higher for blocked actions), confirm-

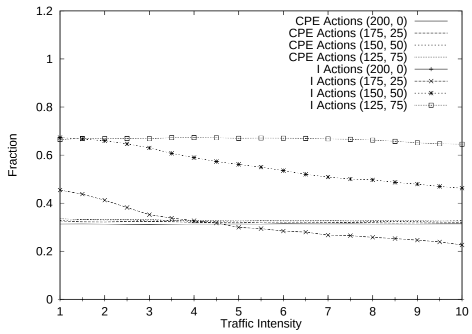

Figure 11. Fractions of CPE and I-actions.

ing 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.

5. Conclusion

In this paper, 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 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 [11] by improving scalability and QoS in both admitting a new request and servicing interactive operations. We then introduced an idealized model to capture several key interaction behaviors. Our experimental results for a scalable batching policy confirm that interactive service can be provided with a minor effect on system scalability. 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.

The additional results, reported in [3], indicate that the CPE mechanisms for continuous and discontinuous operations provide an effective support for adapting QoS to changes in CPE-buffer size, channel capacity, and customers' behavior. In addition, 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. We also introduce in [3] the concept of *active CPE-buffer management*, which is shown to improve customers' QoS when interaction behavior is biased towards a particular type of VCR action.

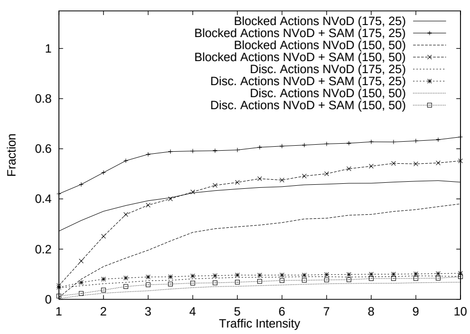

Figure 12. Comparison with SAM.

References

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

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

[3] E. L. Abram-Profeta, *Support for Service Scalability in Video-on-Demand End-Systems*, Ph.D. Thesis, University of Michigan, May 1998.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] K. Keeton, R. H. Katz, "Evaluating video layout strategies for a high-performance storage server," *Multimedia Systems*, Vol. 3, No. 2, pp. 43-52, 1995.

[10] 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*, Vol. 14, No. 6, pp. 1099-1109, August 1996.

[11] 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.

[12] T. D. C. Little, D. Venkatesh, "Popularity-based assignment of movies to storage devices in a video-on-demand system," *Multimedia Systems*, Vol. 2, No. 6, pp. 280-287, 1995.

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