On the Performance of a Multicast Delivery Video-On-Demand Service with Discontinuous VCR Actions

Kevin C. Almeroth
Mostafa H. Ammar

Telecommunications/Networking Systems Group
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
{kevin, ammar}@cc.gatech.edu
(404)894-3292

GIT-CC-94/49

September 27, 1994
Revised: February 15, 1995

Abstract

In most proposed architectures for Video-On-Demand (VOD) systems, the customers are serviced individually by allocating and dedicating a transmission channel and a set of video server resources to each customer. This approach leads to an expensive-to-operate, non-scalable system. We consider a VOD system that uses multicast delivery to service multiple customers with a single set of resources. The use of multicast requires that some interactivity and part of the on-demand nature of the system are sacrificed to achieve scalability and cost-effectiveness. Previous work has considered how VCR actions could be handled in this type of system through the provisioning of set-top box buffering. In this paper we consider a modification of the semantics of the VCR functions in order to avoid this buffering requirement. In particular we consider the provision of *discontinuous* VCR actions in which the duration of the action is specified in integer multiples of a specified time increment. We evaluate the performance of this system through the use of simulation and compare its performance with an individual delivery (unicast) system and a multicast system with continuous VCR actions.

1 Introduction

A Video-On-Demand (VOD) service offers customers a large video library from which they can select a movie to watch at any time they desire. Interactive VOD allows customers to use VCR-style functions such as pause, rewind and fast-forward. The major challenge in providing a VOD service is handling the enormous demand for VHS-quality video in a real-time network with real-time interaction[4]. In most proposed architectures for VOD systems, the preferred approach is to service customers individually by allocating and dedicating a transmission channel and a set of video server resources to each customer. The problem with this approach is that it leads to an expensive-to-operate, non-scalable system. Most VOD trials to-date have been geared toward servicing small groups of customers (see for example [1]). In these systems, as the number of customers increases, so must the number of video server resources and channels.

One of the best ways to deliver information to more than one customer is through the use of multicast communication. Earlier work examined some of the issues in using multicast communication as an information delivery mechanism[7, 8]. The nature of VOD systems logically demands individualized service, whereas with the case of multicast communication, true interactivity and true on-demand behavior is difficult to provide. Systems in which some interactivity and part of the on-demand nature are sacrificed to achieve cost-effectiveness or other objectives are sometimes referred to as *Near* VOD systems[5].

Previous work has addressed the provision of such systems through the use of multicast communication (or batching) in which multiple customers with similar requests are serviced simultaneously[2, 3]. In previous work by the authors of this paper[2], we have considered a system that uses multicast communication to deliver video streams to multiple customers. A limited number of channels are available and customers are allocated channels (after being batched through a slotting mechanism) on a first-come, first-served basis. Customers that cannot be allocated channels are blocked. It is shown that the number of customers that can be serviced in such an environment is significantly greater than for VOD systems using unicast communication. The multicast system allows the customers to issue pause requests which are accommodated through the combined use of set-top box buffering and multicast stream switching. In [3], batching of requests is used and policies for selecting which movies to serve are evaluated. In that system customers that wait too long may renege and one of the objectives in choosing a scheduling policy is to minimize the effect of reneging. The system also allows pause actions which may be serviced through set-top box buffering or through changes among multicast streams.

One of the drawbacks to the way pause requests are handled in the above systems is the need to provide buffering in the set-top box. In this paper we consider a modification of the semantics of the pause function as well as rewind and fast-forward in order to avoid this buffering requirement. In particular we consider the provision of *discontinuous* VCR actions in which the duration of the action is specified in integer multiples of a specified time increment. We evaluate the performance of this system through the use of simulation and compare its performance with that of a unicast system and of a multicast system with continuous VCR actions (as proposed in [2]). The use of discontinuous VCR actions has been discussed in various VOD system proposals[5]. We are not, however, aware of any extensive performance study of such a system and in particular how its performance compares to various other alternatives.

The paper is organized as follows. In Section 2 we discuss the architecture of our Video-On-Demand system. In Section 3, we discuss the operation of a Near VOD system using multicast and discontinuous VCR actions. Section 4 discusses how we evaluate the performance of our system. Section 5 contains numerical results obtained through a simulation of our system. The paper is concluded in Section 6.

2 System Architecture

The architecture of our Video-On-Demand system is shown in Figure 1. The system components include the video server, the network, and the set-top box. Below is a brief description of the functions of each.

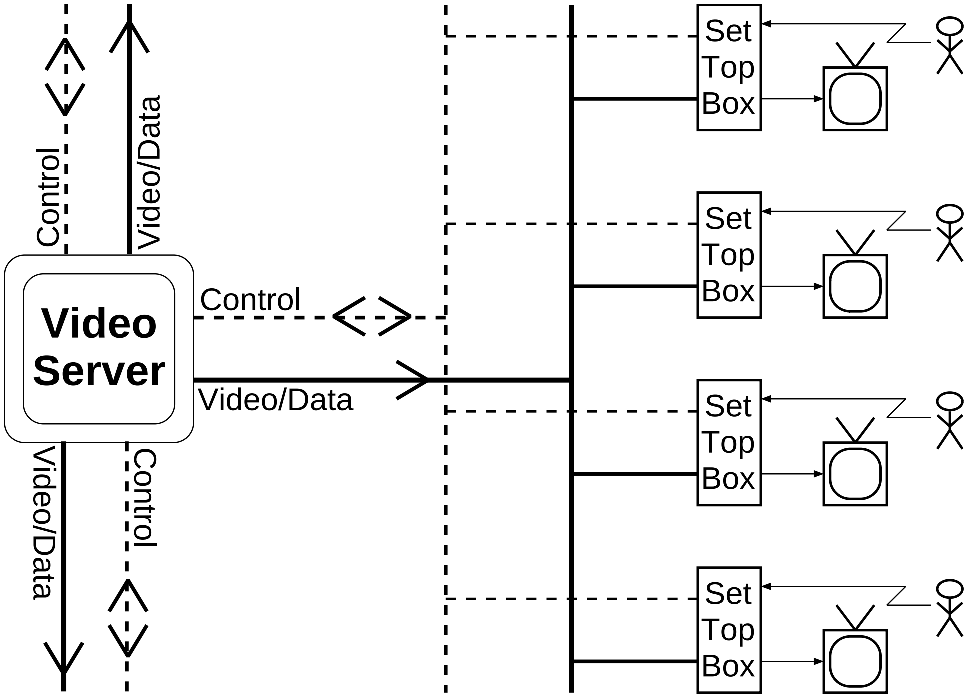

Figure 1: The architecture of our VOD system.

1. **The Video Server**: The video server is responsible for receiving and processing customer requests; and setting up and maintaining video streams which deliver requested movies. The other main function of the video server is to retrieve movie data that is stored on disk, and send it to customers. The video server is a control point for processing and forwarding movie frames. The issues involved in retrieving and sending video frames have been the subject of some recent research (see for example[6]). Furthermore, the ability of the video server to simultaneously support large numbers of streams will depend on the video server architecture. Scalability can be achieved by using arrays of smaller capacity servers.
2. **The Network**: The functions of the network are to transmit video frames from the video server to the customer, and to transmit control messages. The topology that we are using is a star-bus configuration

with the video server at the center of the star. All customers are connected via the bus network so that a video stream sent to one customer can be received by any other customer in the same neighborhood. This architecture facilitates the use of multicast to service multiple customer requests with the same network channel. And finally, the ability of the network to simultaneously support large numbers of streams will depend on the ability of prevailing technology to efficiently use bandwidth for the variety of available media.

3. **The Set-Top Box**: The two classes of functions the set-top box must provide include sending and receiving control messages; and receiving and processing a video stream. The set-top box also receives and processes input from the customer via a remote control. The customer can either request a movie; or during a movie, request VCR-style functions such as pause, rewind, and fast forward. The set-top box will process these signals, and send the appropriate requests upstream to the video server. Part of the responsibility of the set-top box is the playout of a video stream with minimal interruption of service. In fact, *any* interruption in service is considered to be a significant error. Achieving this strict requirement is accomplished by reserving resources at the video server and in the network to guarantee the timely arrival of video frames. Buffering a small number of frames in the set-top box will also help provide continuous playout in the event of short, unexpected delays in the video server or network. Received frames are buffered until just before playout when they are uncompressed with the appropriate hardware. The final component is an inexpensive CPU to control the set-top box operation. An important consideration in the design of the set-top box is its cost. By limiting the complexity and operating requirements, we hope to keep the cost of the set-top box low.

3 System Operation

3.1 Use of Multicast

In our system, movies are made available only at the beginning of slots. The slot duration is on the order of minutes (in our analysis we use the range 1 to 20 minutes). A customer making a request will thus have to wait for, on average, half a slot duration before the movie can start. For short slot durations (say 5 minutes) this should not affect the "on-demand" nature of the system.

At any point in time, customers make requests specifying a movie to watch. A request is sent to the video server, which immediately processes the request and determines if resources are available to service it. Customers are informed through response messages whether their request is accepted or denied. All requests that arrive during the current slot are scheduled or rejected before the end of the slot.

The synchronization of movie start times to a slot boundary allows the server to serve all customers who requested the same movie during a slot using a single set of resources. The video server starts multicasting a movie at the end of a slot to all customers that request the movie during the slot. Multicast groups are formed by having multiple set-top boxes listen to the same channel. Assuming a Frequency Division Multiplexed System, a multicast group is identified by a particular channel and joining a group can be accomplished by tuning to the appropriate frequency. We use the term "multicast communication" because even though every set-top box receives the transmission, not all customers have requested the stream. Only a subset of set-top boxes will receive and process a particular multicast video stream.

Our ability to group multiple movie requests will depend on several factors. The longer the slot duration is, the more likely that several requests for the same movie will arrive during the slot. Of course, long slots detract from the "on-demand" nature of the system. The other factor is the popularity of a particular movie. The system is more likely to be able to group requests for currently popular movies than for a less frequently requested movie. Movie rental statistics[9] and Zipf[10] suggest that a small percentage of the movie offerings will experience the largest request volume. It is for these movies that the benefit of the use of multicast becomes apparent.

3.2 Providing VCR Capability

Once the movie starts, the customer can pause, rewind, or fast forward, the movie. In the system we consider in this paper, the VCR actions can only be specified for durations that are integer multiples of a predetermined *time increment*, τ. For example, if this time increment is 2 minutes, a customer can pause for durations that are $2, 4, 6, \ldots$ minutes long. For rewind and fast forward, the customer can move what he is viewing by periods of $2, 4, 6, \ldots$ minutes in the reverse and forward directions respectively. In our analysis we allow the time increment to range from 30 seconds to 10 minutes. A system which provides this level of interactivity is called Near Video-On-Demand. VCR actions in such a system are "discontinuous", as opposed to a continuous system which allows VCR actions of any time length.

3.2.1 Actual and Logical Start Times

To explain the effect of discontinuous VCR actions on the system, we define with respect to each customer viewing a movie, the *actual* and *logical* movie start times. The actual movie start time defines the time the movie started playing. The logical movie start time is defined as the current time minus the duration of the portion of the movie from its beginning until the point at which the customer is watching. Initially, the logical and actual start times are the same. At the conclusion of a pause for n time increments, the n increments are added to the logical start time. A fast forward (or rewind) of n time increments subtracts (or adds) n time increments from (to) the logical start time.

Figure 2 illustrates how a customer VCR action causes a change in the logical start time. In Figure 2 we plot the customer's position in the movie (represented by the solid line) as a function of real time. A pause of $n\tau$ time units causes the customer's position to remain constant for that length of time. A rewind or fast forward action of $n\tau$ time units causes the customer's position to jump instantaneously up or down, respectively, by $n\tau$ time units. At the conclusion of a VCR action, the progress of a customer's position continues as before. By extrapolating the line that represents the customer's viewing progress after the VCR action (the dashed line) to meet the horizontal axis, we can determine the logical start time relative to the customer. It should be emphasized that the logical start time concept is with respect to a customer. Video streams, once started by the video server, are typically not affected by customer VCR actions (except in some circumstances as described below).

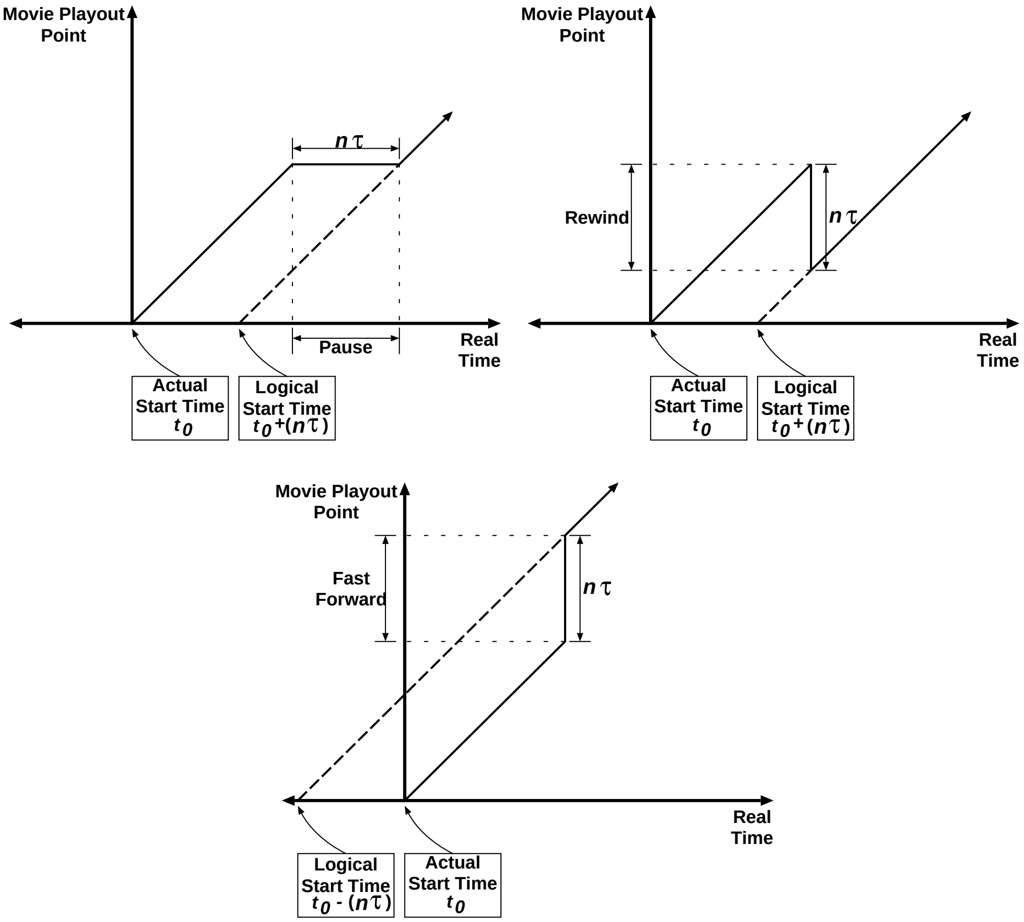

Figure 2: Effect of each VCR action on a movie's logical start time.

3.2.2 Responding to VCR Actions

Because multicast communication is used and multiple customers may be watching the same video stream, it is not always feasible to modify the video stream as a result of the VCR action. As customers issue VCR actions, the logical start times of the movies they are watching change. The idea behind providing uninterrupted service to a customer is to match the customer with a video stream carrying an appropriate movie that was started at the customer's logical start time. Multicasting is of benefit if, at any point in time, multiple customers watching the same movie possess the same logical start time. The use of time increments has the effect of increasing this likelihood.

When a customer issues a VCR action, a message is sent to the video server with a new logical start time. When the video server receives a multicast group change request it determines the number of customers watching the same movie at the original logical start time and at the new logical start time. Using this information the video server determines if the VCR action request can be granted. The procedure used by the video server is as follows:

1. The video server must first determine if the resources exist to grant the VCR action request. A request can be granted if one of the following two conditions can be met:

   a. If another stream exists which is showing the same movie at the new logical start time.
   b. If no such stream exists, but resources are available which can be allocated for the creation of the appropriate stream.

2. If the VCR action request can be satisfied, the customer who performed the VCR action is moved to the existing or newly created stream. Depending on the number of customers left in the original stream, one of two scenarios is possible:

   a. If the customer who moved to the new stream was the only one watching the old stream, the stream is no longer needed and can be deallocated. These resources can be used to satisfy other initial requests or VCR action requests which require a new channel.
   b. However, if there are other customers watching the stream, then no deallocation can occur.

It should be noted that there is always the danger that a request to change to a different video stream cannot be granted. This will happen when there are no existing video streams to satisfy the request, and no idle channels are available. We consider this type of VCR action blocking to be unacceptable and would therefore like to design a system where the probability of this blocking is extremely low. Our approach is to trade-off slightly higher initial movie request blocking with lower VCR action blocking. This is accomplished by reserving a number of “emergency” channels. These are not available to satisfy initial requests and are used exclusively to serve requests to change video streams that would otherwise be blocked. In the extreme case when no emergency channels are available, VCR actions that cannot be accommodated are blocked. A blocked VCR action simply means that the action does not occur, and playout of the movie continues. Alternatively, the customer can be queued for service at a later time. We do not consider this latter option which has been analyzed in [3].

3.3 Video Server Operation

The video server receives two types of requests from customers: (1) initial requests for movies, and (2) multicast group change requests. For initial requests, the video server must determine whether or not a customer request can be satisfied. The video server uses information about system load and the following procedure to make the determination.

1. If a channel has already been reserved for the requested movie at the requested time; the video server simply adds this customer to the already existing group; and sends a “request accepted” message.
2. If a group does not already exist, and there is a free channel; the video server creates a new group; allocates a free channel to the group; and sends a “request accepted” message.
3. If the group does not exist, and there are no free channels; the video server sends a “request denied” message. We use a blocking model and assume that a customer who is denied service will not try again.

Figure 3 shows the flow of request and response messages sent when a customer makes a movie request. Note that the video server performs straight-forward First Come, First Serve scheduling. That is, no request is denied if a channel is available (free or assigned to the same movie) at the requested time. Other scheduling disciplines are possible, and some have been investigated in [3].

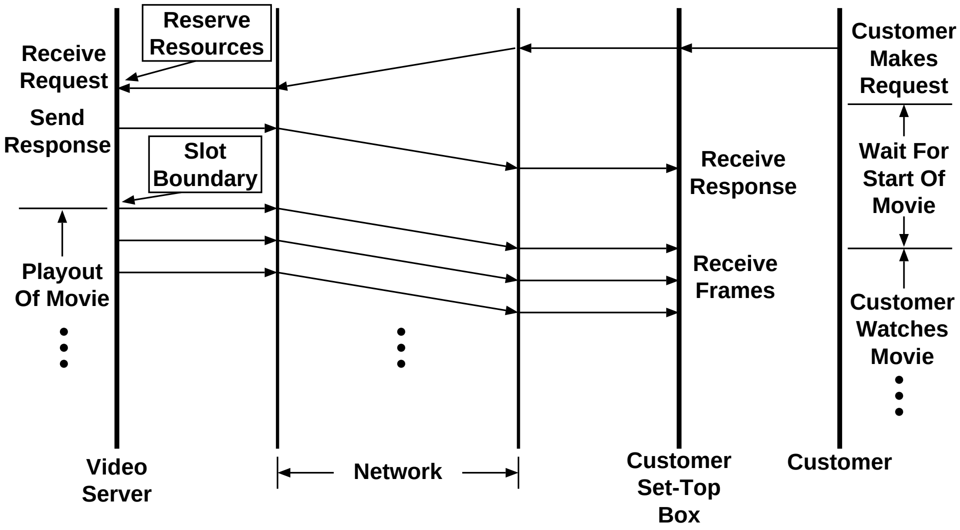

Figure 3: The flow of a customer request and video server response.

The second type of request that the video server must handle is customer multicast group change requests. A change request is sent by a set-top box when a customer uses a VCR function. Upon receipt of a change request, the video server first determines which multicast group the customer is in and the number of listeners in the group. Next, given the duration of the VCR action, the video server computes a new logical slot time. The video server then uses the procedure described in Section 3.2.2 to determine if a stream exists with the same logical slot time. Figure 4 shows the flow of change request and response messages that flow between the system components when a customer uses a VCR action.

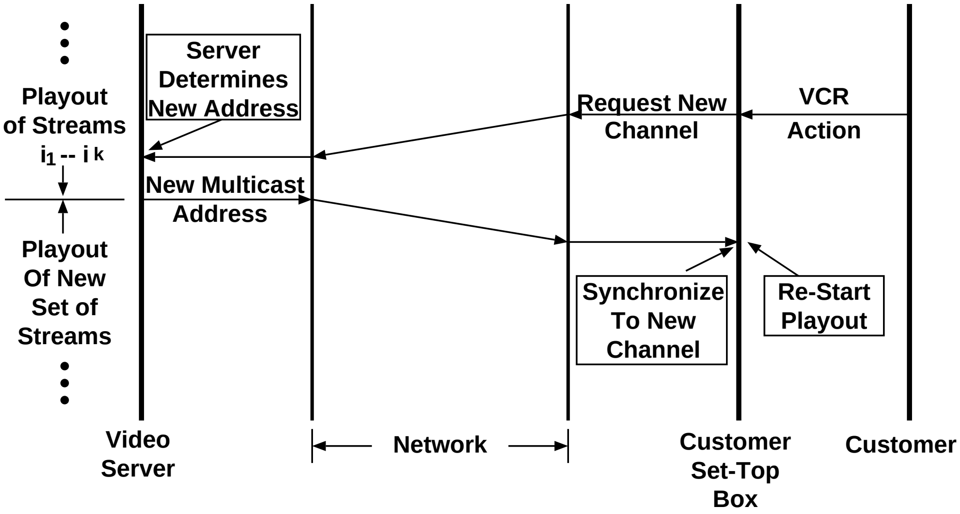

Figure 4: The system actions which take place when a customer initiates a VCR action.

3.4 Set-Top Box Operation

The following describes what the set-top box needs to do in response to a request generated by a customer's VCR action.

- **Rewind**. The customer uses a remote control and requests that the movie be rewound a number of time increments. The set-top box sends a request to the video server with this information. The video server responds with a new channel or an "action blocked" message. The set-top box simply tunes to the new channel, and continues playout of newly received frames. If the action is blocked, playout continues.
- **Fast Forward**. The operation of the fast forward function is the same as rewind, except that the request to change groups is forward in the video stream.
- **Pause**. The pause function is slightly different. When a customer starts a pause, a request is immediately sent to the video server to change to a new group. A pause is treated just like a rewind at the video server. The video server responds with a new channel to tune to. The set-top box immediately tunes to the new channel, and starts receiving video frames that have already been displayed. These frames are discarded until the set-top box receives the frame that was being displayed when the pause action was started. When this frame is received, playout continues. It takes the pause duration for the new video stream to catch up to the frame when the pause occurred.

At this point, it is appropriate to mention the delays associated with initiating a VCR action. There will be a small delay between the time an action is requested, and the time the action actually happens. This delay is equal to the propagation delay of sending the request; plus the time to process the request; plus the propagation delay of sending the response. Considering that today's VCRs also have delays due to the physical limitations of starting and stopping the tape, this request processing time should not be noticeable.

A set-top box that operates as described above can be built using either current technology, or technology that will soon be available. Since this set-top box only allows discontinuous VCR functions, it is less expensive than a set-top box that uses buffering to provide continuous VCR functions as discussed in [2].

4 Performance Evaluation

In order to analyze and compare the performance of our proposed Video-On-Demand system, we developed simulations to model the operation of unicast and multicast VOD services. Using a set of parameters, the simulation can be configured for different sized systems servicing customers with varying behavior patterns.

For a given set of parameters, a simulation models the operation of a VOD system during the six "prime time" hours of highest expected system utilization. "Prime time" includes the evening hours which typically experience a majority of the day's movie requests[5]. The simulation does not consider any requests that occured just before or just after the simulated period. We make this assumption because the number of requests outside the prime time period is expected to be much lower. However, even though we do not consider requests which arrive after the simulation period, the simulation must continue to run until all requests which occured just before the end of the simulation period have finished. In addition to the six hour prime time period, a simulation will run for a period of time equal to the length of a movie.

4.1 System Parameters

The system parameters used to understand the performance of the VOD systems include the following:

1. **Number of Customers Making Requests**. This is the total number of customers who will make requests during the simulation period. Each customer makes exactly one request, and does not re-try if the request is denied.
2. **Total Number of Channels**. This is the total number of logical channels available for playing video streams. A logical channel includes both bandwidth resources in the network, and playout resources at the video server.
3. **Number of Emergency Channels**. This is the number of channels reserved for preventing VCR actions from being blocked. These channels cannot be used to satisfy initial customer requests.
4. **Slot Length**. This defines the length of time between movie starts. All requests for the same movie that arrive after the end of the last slot, and before the end of the current slot are held and serviced together at the end of the current slot. The slot length also defines the maximum amount of time a customer has to wait before a successfully scheduled movie starts playing.
5. **Time Increment for VCR Action**. This specifies the time increment for any VCR action. The action duration is measured in seconds. For most of the tests, the time increment is made the same as the slot length. In one set of tests, we look at using smaller time increments.
6. **Number of Movies**. This is the number of movies offered to the customer.
7. **Length of Movie**. The movie length specifies how long the playout time of any movie is. All movies are assumed to be the same length.

The rest of the system parameters are used to model the behavior of each customer. They include the following:

8. **Time of Request**. This is the time during the simulation period that each customer makes a request. A request is equally likely to be for any slot.
9. **Movie Selection**. We assume the probability that a request is for movie i is given by q_i. The q_i's are assumed to follow Zipf's Distribution[10] i.e., if $q_1 \geq \ldots \geq q_L$ where L is the number of movies, then $q_i = c/i$ where c is a normalizing constant such that $\sum_{i=1}^{L} c/i = 1$.
10. **VCR Action Usage**. Once the movie starts, a customer can affect the playout of the movie by using one of the VCR action buttons. Figure 5 represents the viewing pattern for one customer watching one movie. After an initial request has been made, and the customer starts watching the movie, the customer's use of a VCR function can be modeled as an event arriving periodically, and then being processed. After an action is completed, the customer continues to watch the movie until the next action time. Modeling when an action is made, and the effect of the action is important because it affects when and how often customers change multicast groups. User behavior can be modeled in several different ways. When an action occurs, the type is randomly chosen, and all three are equally likely. The duration of a pause, or the offset length of a rewind or fast forward action is uniformly distributed between 60 and 600 seconds. The actual length of a desired VCR action is then rounded up to the next integer multiple of the time increment. A customer that would like to request a VCR action of duration Y will actually have to request an action of duration $\lceil \frac{Y}{\tau} \rceil \tau$. The difference $\lceil \frac{Y}{\tau} \rceil \tau - Y$ is called the *Undesired VCR Action Length.* The time between the conclusion of one VCR action request and the start of the next is exponentially distributed.

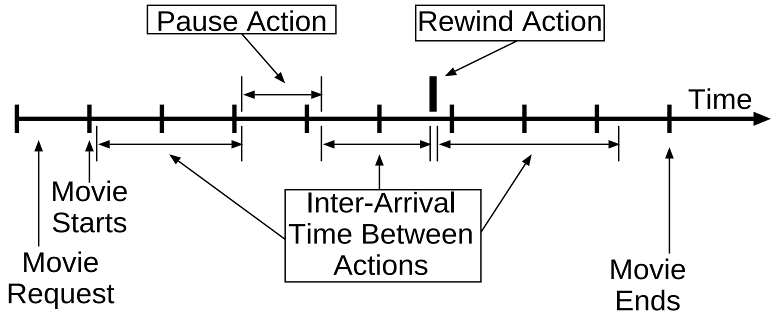

Figure 5: Time line showing movie playout and customer VCR actions.

4.2 Performance Measures

In evaluating and comparing the performance of the simulated Video-On-Demand systems, we used the following three performance measures:

1. **Initial Request Blocking**. This is the percentage of initial requests that were blocked. A request is blocked when there are no free non-emergency channels to allocate to the new movie request.
2. **VCR Action Blocking**. This is the percentage of customer pause, rewind, or fast forward attempts that were blocked because there were no free or emergency channels.
3. **Undesired VCR Action Length**. This is the amount of time that a discontinuous VCR action functions beyond what is desired by the customer. We estimate the desired action length and compare it to the actual, discontinuous action length.

5 Numerical Results

In the results to follow, we consider the effect of a set of parameters on the performance of several VOD systems. Table 1 shows a complete list of parameters studied including the range of values tested, and the nominal values.

Factor	Range Of Values Tested	Nominal Values
Number of Customer Movie Requests	0 to 20000 requests	5000
Total Number Of Channels	100 to 1000 channels	600
Emergency Channels	0 to 10 channels	6
Slot Length	1 to 20 minutes	5
Time Increment for VCR Action	$\frac{1}{2}$ to 10 minutes	5
Number of Movies	50 to 1000 movies	500
Length of Movie	80 to 160 minutes	120
Average Time Between Customer VCR Actions	10 to 120 minutes	60

Table 1: The range of values studied, and the nominal value for each factor.

Section 5.1 reports the results for a multicast VOD system with discontinuous VCR actions and all parameters set to their nominal values. Section 5.2 reports the results of a comparison between a unicast VOD system and a multicast VOD system. In this comparison, eight sets of simulations were run. In each simulation a different parameter was varied over the range of test values. Section 5.3 reports the results of a comparison between three VOD systems. Simulations were run for unicast and multicast VOD systems with varying levels of interactivity.

5.1 Analysis of Nominal System

Tables 2 and 3 show a detailed breakdown of results obtained from a simulation of the nominal system.

Movies	Multicast System Number Accepted	Multicast System Number Rejected	Multicast System Percent Rejected	Avg Group Size	Unicast System Number Accepted	Unicast System Number Rejected	Unicast System Percent Rejected	Avg Group Size
1- 5	1344.4	345.8	20.46	4.397	615.0	1075.2	63.62	1.00
6- 50	1186.2	447.1	27.37	1.520	593.2	1040.0	63.68	1.00
51-500	1150.8	525.7	31.36	1.218	600.5	1076.0	64.18	1.00
1-500	3681.4	1318.6	26.37	1.816	1808.7	3195.5	63.83	1.00

Table 2: Detailed results for initial requests in the nominal system.

Table 2 shows a breakdown of initial requests by groups of movies. Because of the non-uniform popularity of movies, only a few of the most popular movies are needed to represent approximately one-third of all requests. Since more popular movies are selected more often, there is less of a chance that a request for such a movie will be rejected. If all available channels are currently reserved, then only requests for already scheduled movies will be granted. More popular movies are more likely to already be scheduled, and more likely to be requested. This result can also be seen in the average group size which is computed by averaging the number of customers serviced by each channel at the end of each time slot. Since more popular movies are requested more frequently, each reserved channel will be able to service more customers. Each of the most popular movies has an average of more than 4 viewers per video stream. For the least popular set of movies, just slightly more than one customer is serviced per stream. Notice that the group size for the unicast system is always equal to 1.

System	Action	Blocked	Emergency $M \to 0$	$M \to 0$	$1 \to 0$	$M \to M$	$1 \to M$	Total Changes	Percent Blocked
Multicast	Pause	72	126	362	479	996	321	2356	3.035
	Rewind	71	118	323	462	949	288	2211	3.213
	Fast Forward	65	114	327	429	968	304	2208	2.940
	Total	207	358	1013	1370	2914	912	6775	2.940
Unicast	Total	0	0	0	3130	0	0	3130	0.000

Table 3: Detailed results for group changes in the nominal system.

Table 3 shows a breakdown of the types of group changes that occurred during one simulation run. For the multicast system, the results are broken down based on the type of action. Briefly, the types of actions are:

- **Blocked**. A request is blocked because no free or emergency channels were available.

- **Emergency M → 0**. A request changes from a group with more than 1 member to a group that does not exist. An emergency channel is used to satisfy the request.
- **M → 0**. A request changes from a group with more than 1 member to a group that does not exist. A non-emergency channel is available to satisfy the request.
- **1 → 0**. A request changes from a group with only 1 member to a group that does not exist. The video stream is modified to accommodate the sole viewer.
- **M → M**. A request changes from a group with more than 1 member to a group that already exists.
- **1 → M**. A request changes from a group with only 1 member to a group that already exists. This type of transition frees a channel.

5.2 Comparison to a Unicast System

We now consider the effect of various parameters on the initial request blocking probability and VCR action blocking probability of the following two VOD systems:

1. **Unicast Communication with Continuous VCR Functions**. In this system, only one logical channel is allocated per customer; even if two customers request the same movie at the same time. Since customers are each allocated one set of resources, any VCR request may be serviced without VCR Action blocking.
2. **Multicast Communication with Discontinuous VCR Functions**. This proposed VOD system uses time slots and multicast communication to satisfy multiple customer requests with one set of resources. Discontinuous VCR functions are provided, but an action might be blocked if resources are unavailable.

5.2.1 Effect of the Number of Customers Making Requests

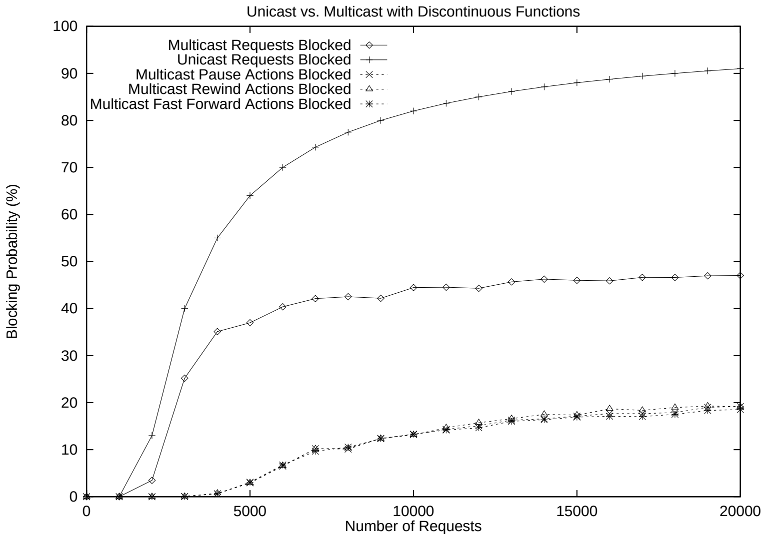

Figure 6: Blocking probabilities as a function of the number of customers making requests.

Figure 6 shows the impact of increasing the number of requests on the blocking probabilities on a multicast system and a unicast system. Both request blocking probabilities start at zero, but as the number of requests increases, the unicast system initial request blocking probability increases more rapidly. The multicast system is much better at handling a large number of customers. More requests mean more chances to service multiple customers with the same logical channel, which means a slower increase in the initial request blocking probability. The VCR actions blocking probability also starts out at zero, but starts to increase slowly after 5000 requests. This is because as the system becomes saturated with requests, fewer non-emergency channels are available to satisfy group changes. Even at high loads, VCR action blocking probability can be kept small by increasing the number of emergency channels.

5.2.2 Effect of the Total Number of Channels

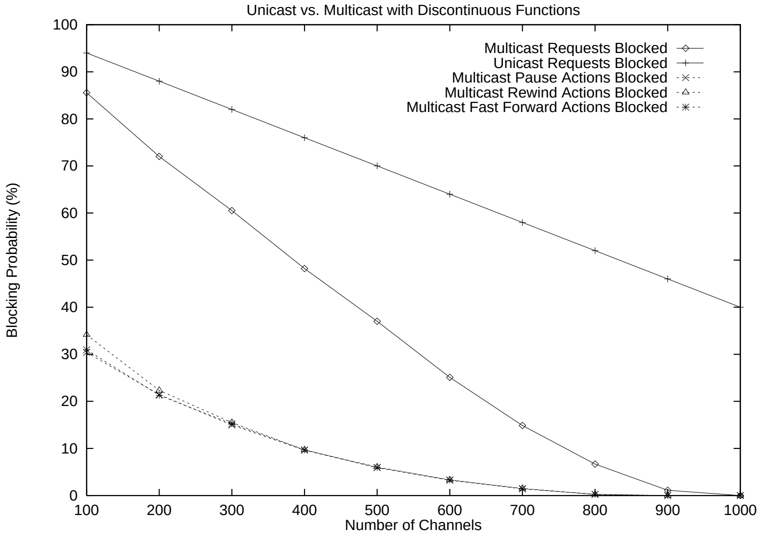

Figure 7: Blocking probabilities as a function of the total number of channels.

Figure 7 shows that as the number of logical channels available is increased, the request blocking probabilities decrease. The requests blocked in the multicast system decrease more rapidly and approach a reasonable level much more quickly. For a system with 5000 requests, 600 channels would produce a initial request blocking probability of about 25%. In the unicast system, requests would be blocked almost 65% of the time. Consider that in this example the multicast system is able to satisfy twice as many customers as the unicast system. These customers are paying customers who reduce the per movie overhead significantly. The VCR action blocking probability starts out relatively high, but as channels are added it becomes acceptable.

5.2.3 Effect of the Number of Emergency Channels

Figure 8 shows the impact of increasing the number of emergency channels on the blocking probabilities. Since the unicast system does not use emergency channels, there is no increase in the blocking probability. For the multicast system, as we increase the number of emergency channels, the VCR action probability decreases, and the initial request blocking probability increases. Since it is important to minimize blocked

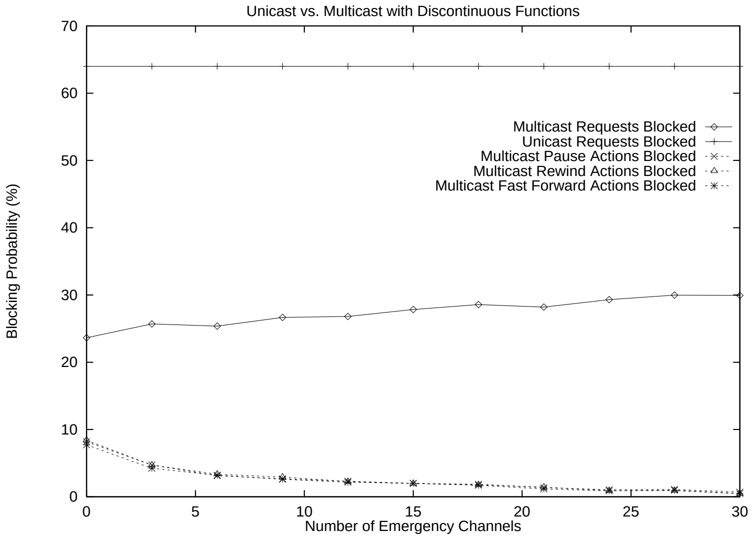

Figure 8: Blocking probabilities as a function of the number of emergency channels.

pause attempts without dramatically increasing initial requests blocked, a good trade-off would be to allocate 6 emergency channels for a system with 600 total channels. This 1% is considered a lower bound, and higher percentages of emergency channels would further reduce the VCR action blocking probability at a cost of only slightly more initial request blocking.

5.2.4 Effect of the Slot Length

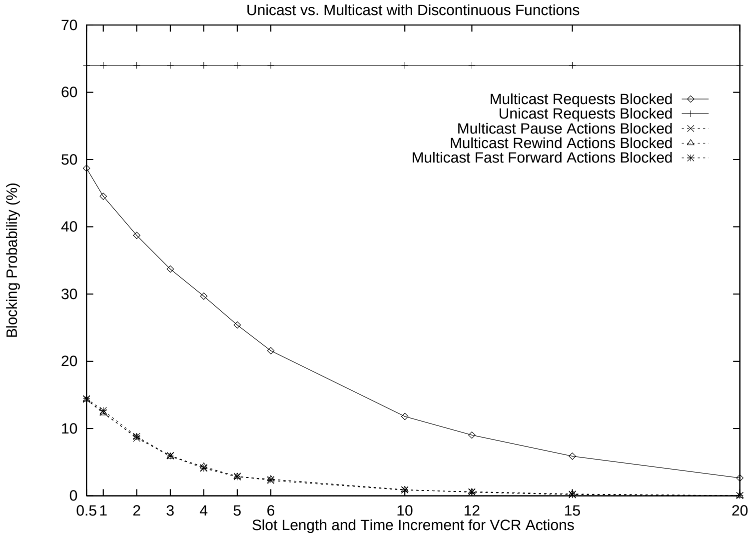

Figure 9: Blocking probabilities as a function of the slot length.

Figure 9 shows the impact of increasing the slot length. It is significant because it shows the trade-off between quick movie starts and the initial request blocking probability. The unicast system is not affected by varying inter-movie times because there are no advantages to delaying a movie. However, in the multicast system, the longer the slot length, the more requests that will arrive during each slot. At the end of the slot, when movies are played, the multicast group size for each movie played will be larger for longer slot lengths. Also, the shorter the slot, the more often group changes will require emergency channels.

5.2.5 Effect of the Number of Movies

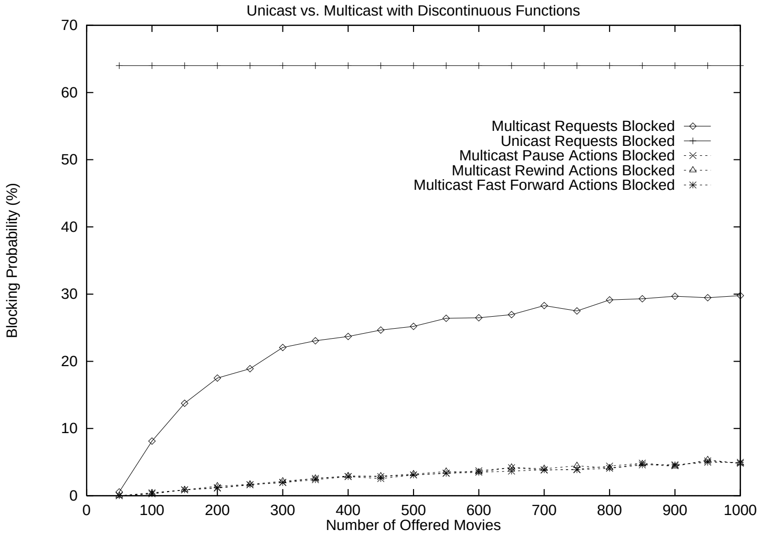

Figure 10: Blocking probabilities as a function of the number of movies.

Figure 10 shows how the number of offered movies affects the blocking probabilities. In the multicast system, the initial request blocking probability increases dramatically at first, and then levels off. This is because in a system with fewer movies, the likelihood that multiple requests will be for the same movie is much greater. In the unicast system, blocking probabilities are independent of the number of movies. The VCR action blocking probability increases slightly because as the number of movies increases, there is less of a chance that an existing video stream will be able to satisfy group change requests.

5.2.6 Effect of the Movie Length

Figure 11 shows that as the movie length increases, the blocking probabilities for both systems also increases. This result is somewhat intuitive because a longer movie requires the same resources, but for a longer period of time. The rate of increase in blocking probability for both systems is about the same, and the VCR action blocking probability increases slightly as additional load is placed on the system.

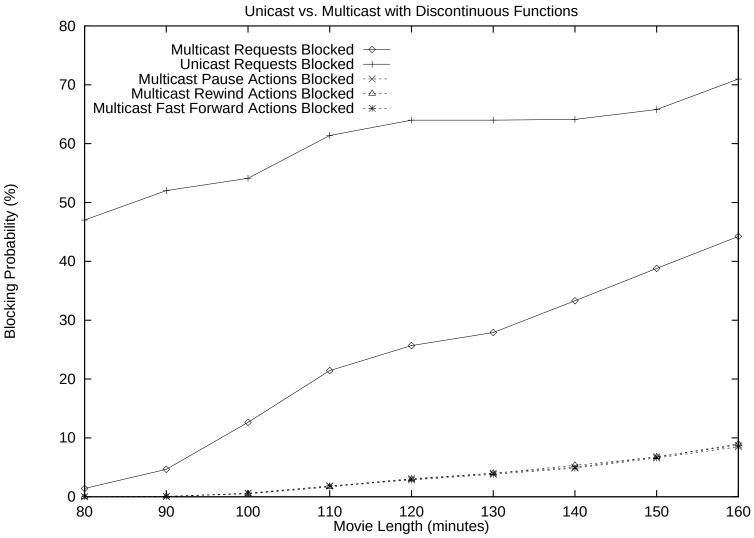

Figure 11: Blocking probabilities as a function of the movie length.

5.2.7 Effect of the Intensity of VCR Actions

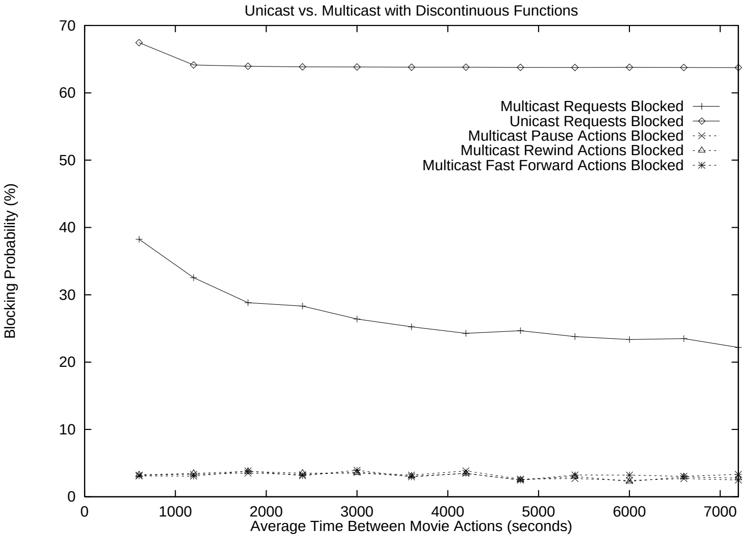

Figure 12: Blocking probabilities as a function of the intensity of customer VCR actions.

Figure 12 shows that as the intensity of customer actions decreases i.e., the time between the conclusion of one VCR action and the initiation of the next, so does the initial request blocking probabilities. Currently, the time between customer actions is exponentially distributed with a mean time measured in seconds. Only when the time between actions is very small is there any significant impact in the unicast system. In the multicast system, fewer group changes mean that there is less of a chance of needing free or emergency channels. Also, since a majority of the VCR actions are pauses, or rewinds, fewer actions means resources do not need to be reserved for additional periods of time. The VCR action blocking probability remains consistently low.

5.2.8 Effect of the Length of the Time Increment for VCR Actions

In this section, we present the results for blocking probabilities for combinations of slot lengths and time increment for VCR actions. Our results are shown for 1% (6), and 10% (60) emergency channels.

With 1% Emergency Channels

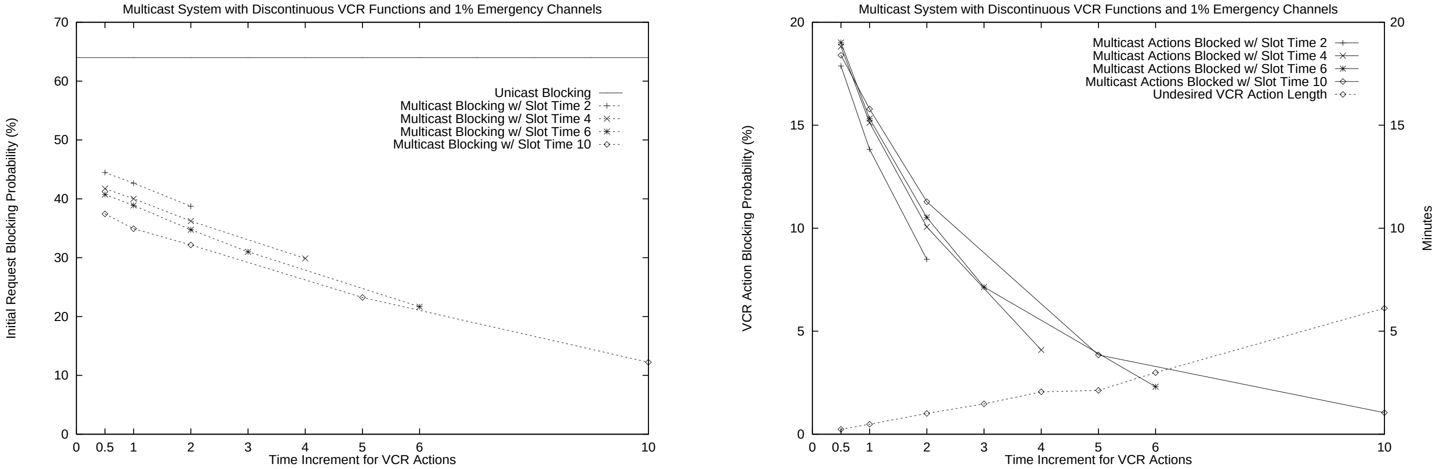

Figure 13: Blocking probabilities as a function of the time increment for VCR actions and with 1% emergency channels.

Figure 13 shows the initial request blocking probabilities for different combinations of slot lengths and time increments for VCR actions. The results show that as the time increment increases, the initial request blocking probability in the multicast system decreases. For the same time increment, but different slot lengths, the initial request blocking decreases slightly for longer slot lengths. The second graph in Figure 13 shows that the VCR action blocking probability decreases significantly for larger time increments. This is achieved at the expense of increased undesired VCR action length.

With 10% Emergency Channels

Figure 14 shows the results of a simulation for a second system. This system has 10% emergency channels. These additional channels increase the initial request blocking probability slightly, but the advantage is a much lower VCR action blocking probability. As more and more channels are reserved for satisfying VCR actions, the lower the probability of VCR action blocking. However, if there are 0% VCR actions blocked, emergency channels are left unused, and wasted.

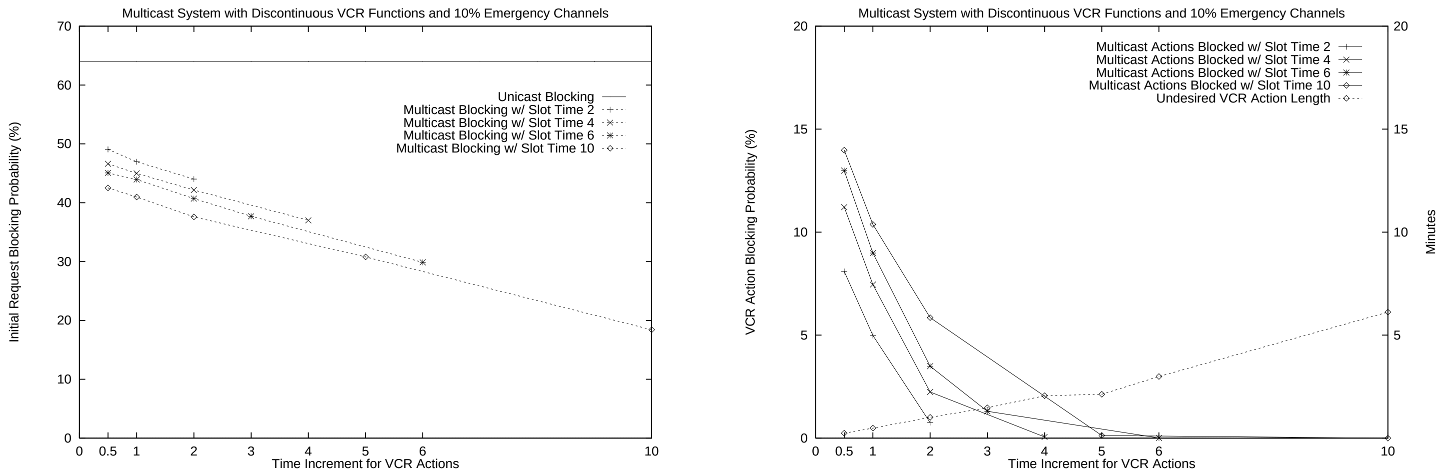

Figure 14: Blocking probabilities as a function of the time increment for VCR actions and with 10% emergency channels.

5.3 Comparison to Continuous Pause Systems

We now turn our attention to comparison of the following systems:

1. **Unicast Communication with Continuous Pause**. This is a limited unicast system which provides a continuous pause function only. Limiting the functions of this system allows us to better compare the three systems.
2. **Multicast Communication with Continuous Pause**. This is a multicast system that provides continuous pause, but at the expense of having a buffer in the set-top box (This system is discussed in [2]).
3. **Multicast Communication with Discontinuous Pause**. This is the system discussed in this paper. However, we only allow discontinuous pause so that we can compare the performance to the other two systems.

As before, we consider the effect of various parameters on the initial request blocking probabilities and the pause blocking probabilities for the three pause-only systems.

5.3.1 Effect of the Number of Customers Making Requests

Figure 15 shows that as the number of customers making requests increases, so does the initial request blocking probabilities in all three systems. In the unicast system, the blocking probability increases much more quickly. For the multicast systems, the blocking probabilities are similar with the discontinuous system having slightly less blocking in most cases. The pause action blocking probability increases for both multicast systems as the number of requests increases. The pause action blocking probability in the discontinuous system is higher because, as mentioned before, not all VCR actions require a group change.

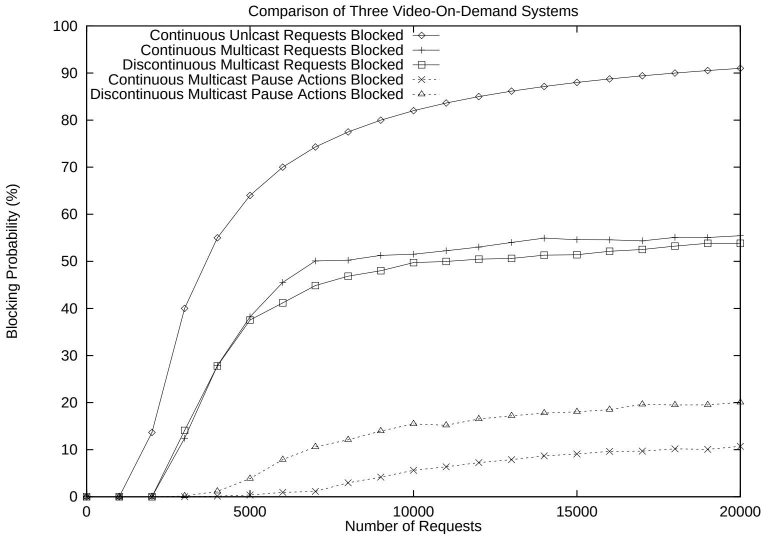

Figure 15: Blocking probabilities as a function of the number of customers making requests.

5.3.2 Effect of the Total Number of Channels

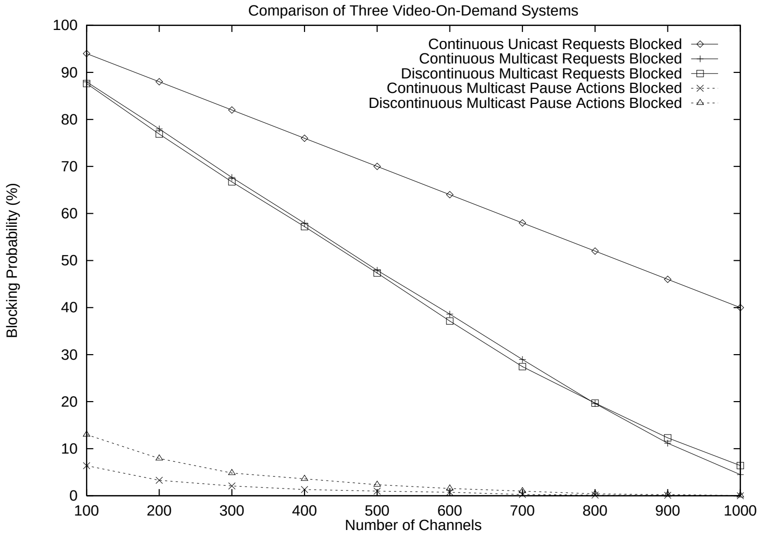

Figure 16: Blocking probabilities as a function of the total number of channels.

Figure 16 shows that as the number of total channels increases, the blocking probabilities in all three systems decreases. The initial request blocking probabilities in the multicast systems decrease more rapidly than in the unicast system. The two multicast systems have similar initial request blocking probabilities. The pause action blocking probability in the discontinuous system starts out higher than in the continuous system. However, as the number of channels increases, the probabilities both become almost zero.

5.3.3 Effect of the Number of Emergency Channels

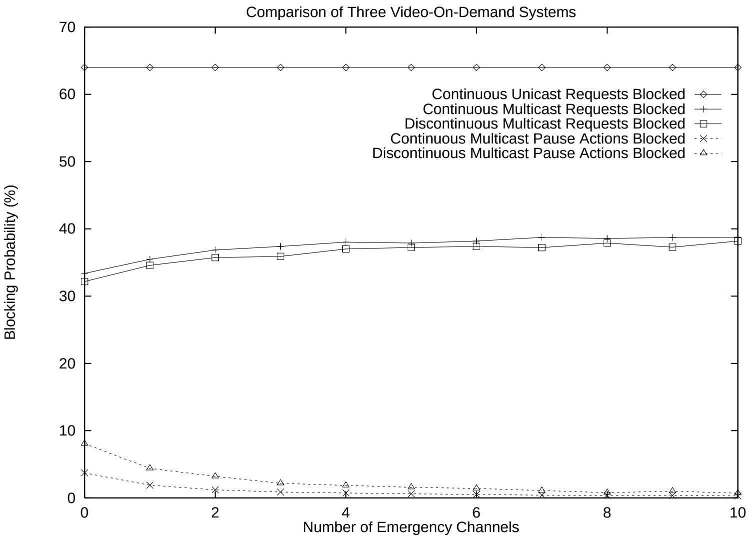

Figure 17: Blocking probabilities as a function of the number of emergency channels.

Figure 16 shows that as the number of emergency channels is increased the pause action blocking probability decreases and becomes almost zero for both systems. The initial request blocking probabilities increase slightly as the number of emergency channels is increased, but the trade-off is very low pause action blocking probabilities.

5.3.4 Effect of the Slot Length

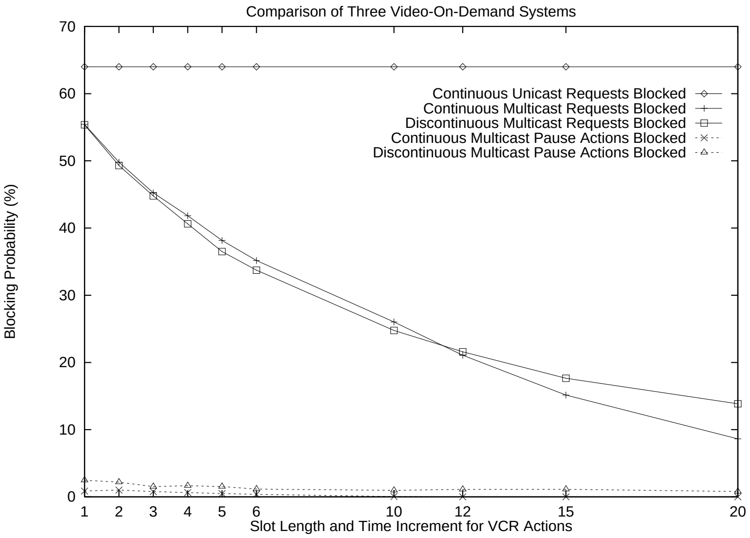

Figure 18: Blocking probabilities as a function of the slot length.

Figure 18 shows that as the slot length increases, the initial request blocking probabilities in both multicast systems decreases rapidly. In the discontinuous system, the initial request blocking probability is slightly higher than in the continuous pause system. At longer slot lengths, the initial request blocking probability is slightly higher in the discontinuous system. The pause action blocking probability for both multicast systems remains consistently low.

6 Concluding Remarks

In this paper, we explored combining the use of multicast delivery and discontinuous VCR actions in a Video-On-Demand system. The operation of such a system was discussed. In particular, we presented mechanisms to handle VCR actions. The performance of the system was investigated through the use of simulation and the effects of various parameters on the performance of the system were studied.

The performance of a multicast delivery system with discontinuous VCR actions was compared to two other systems: a unicast system which provided true VCR actions and a system utilizing multicast delivery and providing only a continuous pause. Table 4 is a summary of our findings.

System	Buffering in the Set-Top Box	On-Demand Nature Of Movies	VCR Actions	Initial Request Blocking	VCR Action Blocking	System Complexity
Unicast w/ Cont. VCR	Not Required[1]	True	Continuous	High	Lowest (0%)	Low
Multicast w/ Cont. Pause	Required	Near	Continuous (Pause Only)	Low	Lower	High
Multicast w/ Disc. VCR	Not Required[1]	Near	Discontinuous	Low	Low	Moderate

Table 4: Summary of the results of the performance analysis of the three multicast systems.

A system utilizing multicast delivery can potentially service more than one customer with a single set of resources, and can therefore service a larger customer population. However, a unicast system is capable of providing a high level of customer interaction. Multicast systems can provide interactivity either through the use of buffering, or by limiting VCR actions to multiples of time increments. Buffering has the disadvantage of adding system complexity and increasing cost. Providing discontinuous VCR functions allows customers to use a wider range of functions and without buffering in the set-top box.

---

[1] Minimal buffering is required to handle jitter and provide continuous playout.

References

[1] J. Allen, B. Heltai, A. Koenig, D. Snow, and J. Watson. VCTV: A video-on-demand market test. *AT&T Technical Journal*, pages 7–14, Jan/Feb 1992.

[2] K. Almeroth and M. Ammar. Providing a scalable, interactive video-on-demand service using multicast communication. In *ICCCN '94*, San Francisco, CA, Sep 1994.

[3] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling policies for an on-demand video server with batching. In *ACM Multimedia '94*, San Francisco, CA, Oct 1994.

[4] A. Gelman, H. Kobrinski, L. Smoot, and S. Weinstein. A store-and-forward architecture for video-on-demand service. In *ICC '91*, 1991.

[5] T. Little and D. Venkatesh. Prospects for interactive video-on-demand. *IEEE Multimedia*, pages 14–23, Fall 1994.

[6] P. Venkat Rangan, Harrick M. Vin, and Srinivas Ramanathan. Designing an on-demand multimedia service. *IEEE Communications Magazine*, pages 56–64, Jul 1992.

[7] J. Wong and M. Ammar. Analysis of broadcast delivery in a videotex system. *IEEE Transactions on Computers*, pages 863–866, Sep 1985.

[8] J. Wong and M. Ammar. Response time performance of videotex systems. *IEEE Journal on Selected Areas in Communications*, pages 1174–1180, Oct 1986.

[9] J. Yoshida. The video-on-demand demand: Opportunities abound, as ditial video becomes a reality. *Electronic Engineering Times*, March 15 1993.

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