

# End-to-End Analysis of Distributed Video-on-Demand Systems

Padmavathi Mundur, *Member, IEEE*, Robert Simon, and Arun K. Sood, *Senior Member, IEEE*

**Abstract**—The focus of the research presented in this paper is the end-to-end analysis of a distributed Video-on-Demand (VoD) system. We analyze the distributed architecture of a VoD system to design global request handling and admission control strategies and evaluate them using global metrics. The performance evaluation methodology developed in this paper helps in determining efficient ways of using all resources in the VoD architecture within the constraints of providing guaranteed high quality service to each request. For instance, our simulation results show that request handling policies based on limited redirection of blocked requests to other resources perform better than load sharing policies. We also show that request handling policies based on redirection have simpler connection establishment semantics than load sharing policies and, therefore, are easily incorporated into reservation or signaling protocols.

**Index Terms**—Distributed Video-on-Demand, end-to-end admission control, performance analysis, resource allocation.

## I. INTRODUCTION

**D**ISTRIBUTED Video-on-Demand (VoD) systems are expected to be one of the most important services supported by the next generation of high-speed networks, video servers, and distributed multimedia file systems. Using a distributed VoD system, a client will be able to request a video from anywhere and at any time. The issue addressed in this paper is the development of a method for analyzing such a VoD system in terms of end-to-end admission control techniques, request handling strategies, and blocking performance. We analyze the performance of a distributed VoD architecture by designing global request handling and admission control strategies and evaluating them using global metrics through analytical modeling or simulation means. We also analyze and model all subsystems—server, network, and client—involved in providing guaranteed services to an individual request in an integrated framework. The objective of this analysis is therefore, the development of a methodology for determining efficient ways of using all resources in the VoD architecture within the constraints of providing guaranteed high quality services to each request.

VoD research over the last decade has focused on analyzing subsystems. In this paper, rather than restricting the analysis to an isolated subsystem the performance of the VoD system is analyzed as an end-to-end system. The need for an end-to-end analysis arises from the fact that for deployable VoD service to a mass market, we need to sustain good quality of service for prolonged periods of time. VoD clients require guarantees on throughput and bounded delays. If there are not enough resources to provide guarantees, the client is *denied* service through admission control. The objective is to maximize the number of clients being served in the VoD system or minimize the number of requests getting blocked or rejected. This requires careful examination of resource usage on all subsystems taken together.

The main contribution of this paper is a performance evaluation methodology developed to find efficient ways of using all resources in the VoD architecture within the constraints of providing guaranteed high quality service to each request. This is particularly significant because a typical VoD architecture consists of hierarchy of servers and network elements with replicated content and resources that global request handling and admission control strategies are a necessity. The relevance of different types of request handling policies under various design considerations such as replicated and distributed video collection, shared and nonblocking remote clusters is determined using an exhaustive simulation. Through our simulation results we show that request handling policies based on limited redirection of blocked requests to other resources perform better than load sharing policies. We also show that request handling policies based on redirection have simpler connection establishment semantics than load sharing policies and therefore are easily incorporated into reservation or signaling protocols.

Previous research in VoD systems has focused predominantly on topics related to video server design, such as disk scheduling, disk striping and video block placement, admission control at the level of disks, and disk groups [2], [4], [9], [10]. The stringent performance requirements for multimedia data transfer are often stated in terms of Quality-of-Service (QoS) parameters such as throughput, network delays, delay jitter, and error rates. The performance guarantees in terms of these QoS parameters at the network are provided using a combination of traffic reshaping using leaky bucket or token bucket controllers, scheduling schemes, and a signaling mechanism. A flow or a channel with known QoS requirements is set up by using protocols such as Resource Reservation Protocol (RSVP) to reserve network resources along the request path on the Internet [11]. Admission control is enforced prior to resource reservation to make sure

Manuscript received December 7, 2000; revised July 23, 2002. The associate editor coordinating the review of this paper and approving it for publication was Dr. Chung-Sheng Li.

P. Mundur is with the Department of Computer Science and Electrical Engineering, University of Maryland-Baltimore County, Baltimore, MD 21250 USA (e-mail: pmundur@cs.umbc.edu).

R. Simon and A. K. Sood are with the Department of Computer Science, George Mason University, Fairfax, VA 22030 USA (e-mail: simon@cs.gmu.edu; asood@cs.gmu.edu).

Digital Object Identifier 10.1109/TMM.2003.819757

that the QoS requirements can be met on the routing path of an arriving request. Many different network scheduling disciplines have been proposed that provide guarantees on different QoS parameters mentioned before. In Section II-A, we show in our integrated admission control model, the use of Weighted Fair Queueing (WFQ) [7], [8] as the scheduling scheme that guarantees reserved rates and bounded delays.

## II. HIERARCHICAL VoD ARCHITECTURE

A typical VoD architecture consists of three critical subsystems: single or clusters of video servers, high speed wide-area and local distribution networks, and user populations. Proposed hierarchical VoD system architecture consists of local and remote sites. Each site is characterized by a cluster of video servers. The video servers deliver high quality digitized multimedia data to clients over local distribution networks from local sites or over high speed networks from remote sites. Set-top boxes at the client site provide decoding and display functionality, in addition to providing buffers for periodically delivered video segments from the video servers. A typical organization of local and remote clusters in reference to a single user population is shown in Fig. 1.

Each local cluster is dedicated to a single user population, referred to as the *reference* user population. The local cluster may store a complete or a partial set of videos from the video collection. Each video is stored on a server in the cluster and video segments are periodically delivered from that server in response to a client's request for that video. The most popular videos are replicated and stored on different servers in the local cluster. The service from the local cluster is provided over a local distribution network, such as an ATM LAN, HFC, or xDSL. It is assumed that there are sufficient network resources at the local distribution network to deliver videos to the clients and that there is no resource contention on the local distribution network. This is a reasonable assumption, because the local cluster acts as a neighborhood cluster of servers and as the overall VoD user population grows, more new local clusters are added. When a new neighborhood local cluster is added to the distributed VoD system, the network capacities can be sized to match the reference user population for that neighborhood. The requests originating from a reference user population are best served by its own local site because of the absence of network contention. However, clusters of servers have limited bandwidth and therefore reject requests when there are not enough resources to serve them. The local cluster of servers administers admission control tests before accepting new requests. The remote site may be archival in nature, providing a permanent repository for all videos, or they may act as replicated servers such as mirrored sites. Remote servers also provide video delivery service over high speed networks. There is resource contention on the high speed networks connecting to remote sites. The remote sites may provide service to many user populations.

A distributed system of this magnitude lends itself to hierarchies of neighborhoods, regions, and so on. The hierarchy of clusters of servers and networks results in a scalable VoD system

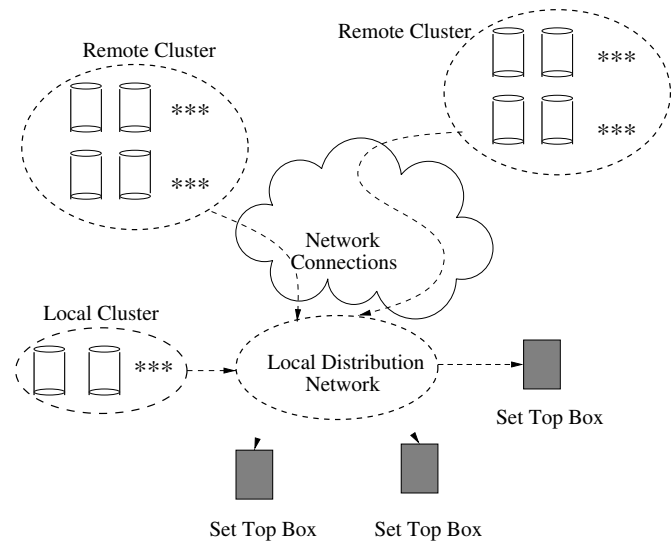


Fig. 1. Hierarchical VoD architecture.

by providing multiple resources to the user population; if a request cannot be served from the local site, it may be directed to other remote sites. In this architecture, the requests are generated by the reference user population. Each request is assigned its own video data stream from either the local or the remote server, so that there is no batching among request streams in the servers. Each video request is served for the duration of the video playback. Though we do not model interactivity, there is nothing in the architecture or in the ensuing analysis that will prevent extending the analysis easily to include VCR-like functions.

### A. Server and Network Model

Research in video server design [4] demonstrates that servers based on RAID-style disk arrays are appropriate because of their retrieval efficiency as well as cost effectiveness. The proposed server model consists of a storage architecture based on high capacity and high bandwidth RAID-5 storage. Using such a video server, the data is transferred to the client periodically; that is, the requests are processed in *rounds* using a disk scheduling algorithm such as Scan EDF [9] or Grouped Sweeping Scheme [10]. The flexibility of implementing any disk scheduling scheme within a round gives the server model a high level of abstraction and makes it general purpose.

The server retrieval and the network transfer are analyzed in an integrated framework shown in Fig. 2. The size of the client set-top box buffer determines the data retrieved at the server. The data retrieved at the server is sent over the network to the client in each round using a *double buffer scheme*. The round length is determined by the duration of the data in the client's buffer played back at a known rate called the playback rate. Starvation at the client buffer is prevented by delivering the new data before the end of the round length. Use of double buffer in a video server has been described before [2]. In the double buffer (see Fig. 2), as the first buffer is being filled from the disk, the second buffer is being emptied into the network and buffer roles are switched at the end of each round. The advantage of using this scheme is that the order in which requests are served from

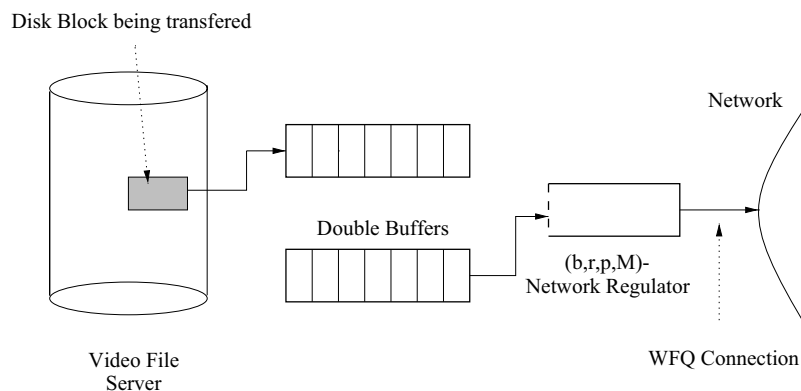


Fig. 2. Double buffer at the server with fair queuing network.

the disk is not important in the first buffer. However, the request data is injected into the network from the second buffer in a constant order.

Serving requests from remote clusters involves data transfer over the network. The network context for the proposed analysis is a high speed, packet switching network along the lines of the Internet Engineering Task Force (IETF)'s *int-serv* and *diff-serv* architectures [1]. The three main components of the network model are a (b r p M) regulator, scheduling scheme like WFQ, and a reservation protocol such as RSVP.

In the proposed network model, the double-buffer at the server acts as the packet source for the *traffic regulator*. It is assumed that a (b r p M) regulator [3] exists at the server. Each request is modeled as a flow passing through this regulator into the network. The data retrieved at the server for each request is packetized instantaneously and fed into the regulator. The regulator monitors the flow of traffic into the network. The regulator guarantees that in any interval  $x \geq 0$ , the number of bytes appearing at its output is given by  $\min(M + px, b + rx)$ , where  $b$  is the token bucket size, and  $r$  is the token accumulation rate. The parameters  $b$  and  $r$  control burstiness of the flow into the network.  $p$  and  $M$  are hardware parameters based on the network interface to the server;  $p$  is the peak rate of the network interface card, and  $M$  is the maximum packet size for the flow. Parameter  $p$  defaults to infinity if unknown but is usually assumed to be greater than  $r$ .

In the network model, WFQ scheduling scheme is assumed to be implemented in the network. In [7], [8], it is shown that WFQ provides a firm per-packet end-to-end delay bound on a per-link and per-routing path and ensures that all transmitted packets will be able to meet this bound. The WFQ delay bound is used in the proposed analysis and inverted to find reserved rates at the network in the next section. In addition to a WFQ scheduling scheme, we assume the use of a reservation protocol similar to RSVP [11] to reserve resources along the path of the requests. In our VoD architecture, connecting to the remote clusters will require such a protocol for providing guaranteed services. The RSVP protocol uses two classes of messages: PATH and RESV. PATH messages carry the traffic specifications of the flow from the sender to the receiver. The receiver responds with the actual reservation request using RESV messages. The RESV message is used to check resource availability at each router based on admission control policies.

### B. Analytical Model

In this section, we develop an integrated analysis by combining the retrieval mechanism at the server with the rate-based transfer mechanism in the network. This integrated framework is critical to our proposed end-to-end analysis of the VoD system. We derive admission control conditions at the server and the network, both of which have to be satisfied for a remote request to get admitted into the system. This end-to-end admission control model is later implemented in the simulation. The model parameters are shown in Table I.

At the server, a retrieval block for each request is retrieved into the double buffer in each round. The retrieval block  $B_r$  is played back at a rate  $R_p$  at the client site. The round length is determined as the playback duration of the retrieval block at the client site. The round length,  $T_{round}$ , for the server is determined by

$$T_{round} = \frac{B_r}{R_p}. \quad (1)$$

As illustrated in Fig. 2, the size of the buffer in the double buffer scheme corresponds to the total data retrieved in a round length for a maximum number of  $n$  requests. The maximum number of requests,  $n$ , serviceable at the server is derived as follows. Let  $R_d$  be the overall disk bandwidth. Then the time to transfer the amount of data retrieved per round is bounded by the round length, that is,  $((nB_r)/(R_d)) \leq T_{round} = B_r/R_p$ . The maximum number of requests that can be served at the server,  $n$ , is  $n \leq \lfloor R_d/R_p \rfloor$ .

An admission control test at the server determines if the arriving request can be admitted into the system. The server is able to serve a maximum of  $n$  requests in a round. Let  $n_c$  be the number of currently accepted requests at the server. The request will be admitted into the system if the following condition holds:

$$n_c + 1 \leq n. \quad (2)$$

For video requests which must use a network connection, the minimum reserved rate,  $R_r$ , at the network required to prevent starvation at the client site is given by

$$R_r \geq \frac{B_r}{T_{round}} = R_p. \quad (3)$$

TABLE I  
MODEL PARAMETERS FOR INTEGRATED ADMISSION CONTROL

$B_r$	Retrieval block size
$R_p$	Client request playback rate
$R_d$	Overall disk bandwidth
$R_r$	Network reserved rate
$T_{round}$	Round length
$n$	Max. requests at the server
$n_c$	Num. of current requests
$D_{max}$	Max. bounded end-to-end delay

The reserved bandwidth cannot be greater than the overall bandwidth of each link or hop on a routing path with  $J$  links,  $R_r \leq A_j$ , for  $j \in [1, J]$  where  $A_j$  is the overall bandwidth on  $j^{th}$  link. Therefore, the bounds on reserved rate at the network,  $R_r$ , are  $R_p \leq R_r \leq A_j$  for  $j \in [1, J]$ .

By using a (b r p M) regulator and a minimum service level of  $R_r$  on a routing path with  $J$  links, the maximum end-to-end delay bound  $D_{max}$  is given by [3]

$$D_{max} = \frac{(b-M)(p-R_r)}{R_r(p-r)} + \frac{(M+C_{tot})}{R_r} + D_{tot} \quad \text{if } R_r < p \quad (4)$$

$$D_{max} = \frac{(M+C_{tot})}{p} + D_{tot} \quad \text{otherwise} \quad (5)$$

where  $C_{tot} = \sum_{j=1}^J C_j$  and  $D_{tot} = \sum_{j=1}^J D_j$ .  $C_j$  and  $D_j$  are error terms originating from the router's approximation of the perfect fluid model [3]. The errors refer to the delay introduced at the router: for instance, a packet arrives but just misses its turn to be scheduled at a router or when a packet has to wait for the transfer of the packet in progress at all routers along the path. The terms  $C_{tot}$  and  $D_{tot}$  represent the summation of the  $C_j$  and  $D_j$  error terms at each link, respectively.

Since  $R_r$  is almost always less than  $p$ , the peak rate of the network interface card, only the case where  $R_r < p$  in (4) needs to be considered. In the proposed model, at the (b r p M) regulator the depth of the token bucket,  $b$ , is the same as the retrieval block per request per round,  $B_r$ . The rate at the token bucket is equal to the reserved rate at the network,  $r = R_r$ . After substituting for  $b$  and  $r$  in (4) and some simplification, the maximum end-to-end delay is

$$D_{max} = \frac{B_r}{R_r} + \frac{C_{tot}}{R_r} + D_{tot}. \quad (6)$$

The values for  $C_j$  and  $D_j$  depend on the scheduling mechanism implemented at the network. Using WFQ, the maximum end-to-end delay bound per packet for a flow with a guaranteed rate of  $R_r$  is given by [7], [8]

$$D_{max} = \frac{B_r}{R_r} + \frac{(J-1)M}{R_r} + \sum_{j=1}^J \frac{M_{max}}{A_j} \quad (7)$$

where  $M_{max}$  is the maximum size of a packet permitted in the network (MTU).

Using (7),  $R_r$  is derived corresponding to an input delay bound  $D_{max}$ . This required minimum rate,  $R_r \geq R_p$ , is used

in the admission control test before admitting a new request on the network:

$$R_r = \frac{B_r + (J-1)M}{D_{max} - \sum_{j=1}^J \left( \frac{M_{max}}{A_j} \right)}. \quad (8)$$

Admission control tests for new flow requests are run on a link by link basis over the routing path. Each router along the routing path will administer these tests to ensure that there are enough available resources. If enough resources are available, they are reserved for the new flow using the signaling protocol. The admission control condition for setting up a new flow is as follows. Suppose that link  $j$  has currently accepted  $T$  flows  $1, 2, \dots, T$ . The  $i^{th}$  flow has a reserved rate of  $R_{r_i}$  computed using (8). If the *new* flow requests a rate of  $R_{r_{new}}$ , it will be admitted only if the following relation holds:

$$\sum_{i=1}^T R_{r_i} + R_{r_{new}} \leq A_j. \quad (9)$$

### III. PERFORMANCE EVALUATION

We next show how to evaluate the performance of our VoD architecture. There are three main objectives. First, we want to quantify the advantages and limitations of various policies under varying load conditions using the proposed evaluation metrics. Second, we want to study the utilization of the underlying resource infrastructure under each of these policies shedding light on their performance. Finally, we want to analyze the effect of policies and design considerations on the performance of request classes. The methodology consists of collecting blocking statistics at the VoD system using discrete-event simulation. In all experiments presented later in this section, the simulation period is kept at 10 000 h. In fact, the results of tests with periods higher than 10 000 h match down to the fourth decimal of the test results with 10 000 h. We used confidence interval analysis at 95% level to verify the accuracy of simulation results. These extended results are available in [5].

#### A. Request Handling Policies

We define two classes of request handling policies both of which form part of the core of admission control design: *redirect* and *split-based* policies. Under redirect policies, a blocked request at one resource is simply redirected to other resources in the VoD hierarchy. The basic idea behind the split policies is load sharing and it works well in a dedicated resource environment. Implementing a split policy with no redirection for blocked requests is the simplest, but efficient splits may be difficult to determine, especially in shared resource environments. A policy based on complete or some redirection has higher implementation overhead because of the multiple checks per blocked request.

Both the redirect and the split policies introduce a certain amount of overhead. The overhead in redirect policies is attributed to additional connection setup time because of the multiple checks performed for each blocked request. Our analysis

TABLE II  
SIMULATION DATA

Servers in local cluster	5
Storage capacity per local server	500 GB
Disk transfer rate at local server	1.2 Gbps
Hops to remote cluster 1	3
Hops to remote cluster 2	6
Max. Transmission Unit (MTU)	1500 Bytes
Maximum packet size	1500 Bytes
Network bandwidth	2488 Mbps
End-to-end delay	300 ms
Size of video collection	150
Size of videos in GBytes	2.46 to 4.8
Service time in hours	0.68 to 2.03

here is based on the approach in Neogi *et al.* [6]. As described in the network model, a reservation protocol such as RSVP is implemented on the routers along the request path. Consider  $l_p$  and  $l_r$  to be the packet latency through a router for the PATH and RESV messages.  $l$  denotes connection setup time with the local cluster. Let  $x$  and  $y$  be the number of hops that the two remote clusters are away from the local distribution network and  $x < y$ . The maximum connection establishment time for a request under the redirect policy assuming that the request gets blocked and redirected to remote cluster that is  $x$  hops away and then again redirected to the second remote cluster  $y$  hops away is given by  $T_{red} = l + x(l_p + l_r) + y(l_p + l_r)$ . In a split policy, the maximum connection setup time experienced by a connection is the time trying to connect to the remote cluster  $y$  hops away and is given by  $T_{split} = y(l_p + l_r)$ . Therefore, the connection setup time for both redirect and split is linear in the number of hops.

Propagation delays are ignored in both cases, as is processing delay during connection set up at the receiver. Both of these delays are common to both classes of request handling policies. With the redirect policy, there is a  $x(l_p + l_r) + l$  additional connection setup time overhead. This overhead according to the experimental results shown in [6] is around 40 ms. We estimate the difference in connection establishment delay between redirect and split policy is in the order of 10's of milliseconds. The impact of the control overhead of ongoing sessions on new connection setup time is also shown to be minimal in [6].

### B. Global Performance Metrics

An end-to-end admission control test will determine if the new request can be admitted into the system based on resource availability on all subsystems involved and other policy considerations. A request that gets rejected by the admission control test is said to be *blocked*. The number of blocked requests forms the basis for developing global performance metrics: *overall blocking probability* and *overall blocking rate*.

The overall average request arrival rate at the VoD system is  $\lambda$ . Suppose that the probability of a request for  $i^{th}$  video is  $p_i$ . The arrival rate for  $i^{th}$  video is  $\lambda_i = \lambda p_i$ . The blocking probability,  $b_i$  for the  $i^{th}$  video is defined as the ratio of the

total number of blocked requests to the total number of arrivals at the VoD system for  $i^{th}$  video. The total number of blocked requests is the difference between the total number of arrivals and the total number of requests served at all resources in the VoD system. Overall blocking probability,  $P_{vod}$  is the weighted average over all videos of the blocking probability per  $i^{th}$  video:

$$P_{vod} = \sum_i^V p_i b_i. \quad (10)$$

The overall blocking rate  $BR_{vod}$  is used as a measure of the performance of the VoD system as a whole. It indicates the number of arriving requests at the VoD system that will get blocked per time unit. The overall blocking rate is defined by

$$BR_{vod} = \sum_i^V \lambda_i b_i \quad (11)$$

In addition to the overall blocking metrics, class-based blocking results are useful. The definition of a class may be based on the characteristic of a video such as the popularity or the characteristic of a request such as its playback rate. A video-based definition includes dividing the video collection into classes based on their popularity index, each class containing a subset of the video collection.

### C. Simulation Setup

The simulation was conducted using the CSIM simulation package. The simulated VoD architecture consists of three levels: *local cluster*, *remote cluster1* via *network1*, and *remote cluster2* via *network2* over a set number of hops. The routing paths, *network1* and *network2*, leading to remote cluster1 and remote cluster2 respectively are fixed. Requests are generated by a single user population associated with the local cluster which is labeled as the reference user population. Requests are served by one of three resources: local cluster, remote cluster1, or remote cluster2. The network capacity on the routing paths is dedicated to the reference user population and not shared by other user populations. The remote clusters are archival in nature and store all videos. They also provide direct real-time video delivery service to arriving requests. The network characteristics are based on high speed, integrated services, packet switching network and network related data are shown in Table II along with other details. The reserved rate at the network corresponding to the input delay is calculated using (8).

The video collection consists of 150 videos and the demand for each video depends on the popularity of the video. The Zipf distribution is used to find the probability of choosing  $k^{th}$  video from a collection of  $V$  videos. The probability of choosing the  $k^{th}$  most popular video from a collection of  $V$  videos, is found by  $Z/k$  where  $Z = 1/(1 + 1/2 + 1/3 + \dots + 1/V)$ . Using the Zipf distribution, if the overall arrival rate of requests to the VoD system is  $\lambda$ , the arrival rate for  $k^{th}$  video is determined by  $\lambda_k = \lambda p_k$  where  $p_k = Z/k$ . Arrival rate based on popularity is used to determine the traffic intensity for each video. Traffic intensity defines the number of requests arriving for a video during the service time of that video request and is used as the basis for replication and placement of videos.

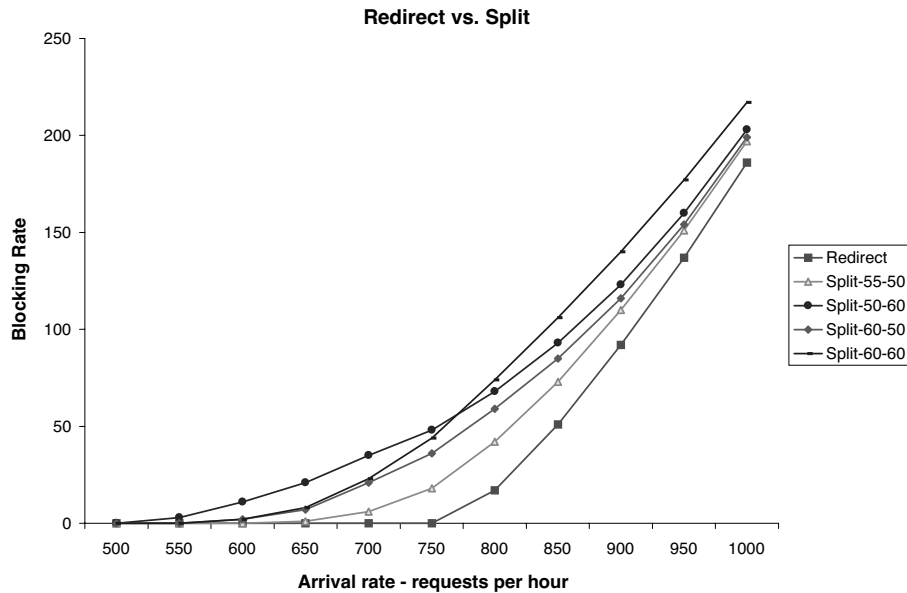


Fig. 3. Blocking rate: redirect and split.

Based on the criteria presented in Section III-A, the following request handling policies are defined for evaluating the performance in this simulation.

- **Redirect**—Blocked requests at the local cluster are redirected to remote cluster1 and blocked requests from remote cluster1 are redirected to remote cluster2.
- **Split-redirect**—The incoming request traffic for videos common to local and remote clusters is first split between local and remote clusters. The remote cluster traffic is again split between remote cluster1 and remote cluster2. For example, a Split-redirect50–60 policy will direct 50% of the incoming request traffic for videos common to both local and remote clusters to local cluster. Of the remote traffic, remote cluster1 will serve 60% and remote cluster2 will serve the remaining 40%. In addition, all blocked requests from local cluster are redirected to remote cluster1. Blocked traffic from remote cluster1 is not redirected to remote cluster2. Several different split values are evaluated.
- **Split**—Similar to Split-redirect, but blocked requests at local cluster are not redirected to remote clusters. Several different split values are evaluated.

#### IV. SIMULATION RESULTS

In this section, we present results of performance evaluation for a single rate playback service. The request playback rate is limited to 8 Mbps. Each server in the local cluster has a disk transfer rate of 1.2 Gbps which translates to 150 simultaneous request streams at 8 Mbps. The request arrival process is assumed to be a Poisson process. The range of mean request arrival rate,  $\lambda$ , is varied between 500 and 1000 requests per hour. We evaluate the performance of the VoD system under two scenarios as enumerated in the following sections. For a replicated video collection scenario, we have all videos from the video collection available on the local cluster. For distributed video

collection scenario, only a partial set of videos from the video collection is available on the local cluster. The requests for the set of videos not available on the local cluster are served by the remote clusters directly.

##### A. Replicated Video Collection

For this scenario, the local cluster consists of five servers and the storage capacity on each server is 500 GBytes with an overall storage availability on the local cluster at 2.5 Terabytes (TBytes). This storage capacity is enough to store the complete video collection on the local cluster. The total storage requirement of the video collection is around 800 GBytes. Additional storage is used to replicate copies of videos as dictated by the replication algorithm (see [5]). The complete video collection is available on the remote clusters as well. The dedicated capacity at the remote cluster with respect to the reference user population is at least as large as the network capacity leading up to them. Therefore, the remote clusters are nonblocking resources. The network bandwidth is assumed to be 2488 Mbps on network1 and network2 and therefore can serve 311 streams of 8 Mbps simultaneously.

Request handling policies evaluated include redirect, split-redirect, and split. Split-based policies are especially relevant in this scenario because it is easy to find an efficient split by simply dividing the incoming traffic in proportion to the stream handling capacity of the local cluster and the network paths. This is possible because the local and remote clusters are acting as replicated clusters with all videos available on them so that a video request can be serviced at any of the three resources.

*System Performance:* The purpose of this experiment is to test the performance of the VoD system using different request handling policies. We have three primary objectives for this experiment. The first objective is to evaluate the general performance of the three main request handling policies, redirect, split-redirect, and split policies under varying load conditions by keeping the VoD system size fixed. The second objective is to evaluate interactions between various split policies. The third

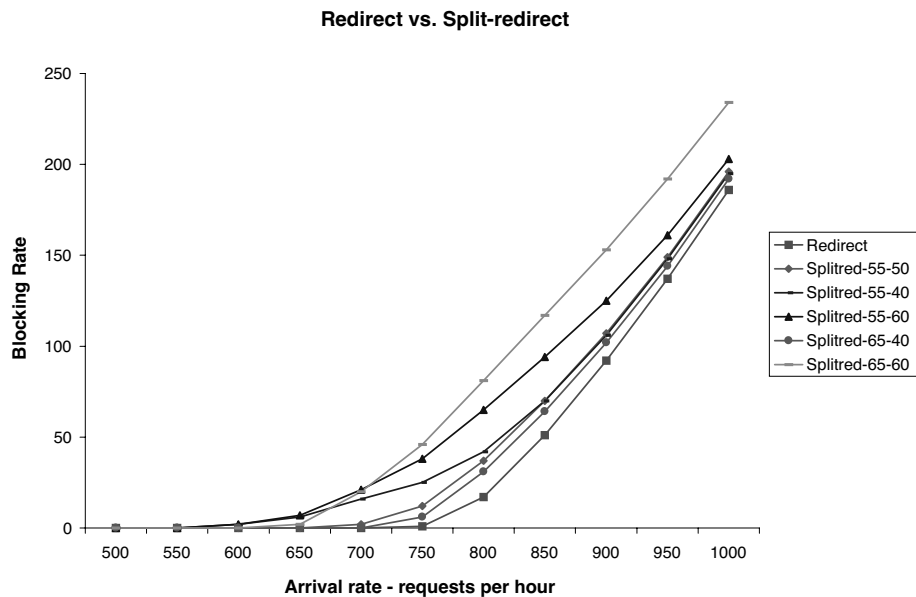


Fig. 4. Blocking rate: redirect and split-redirect.

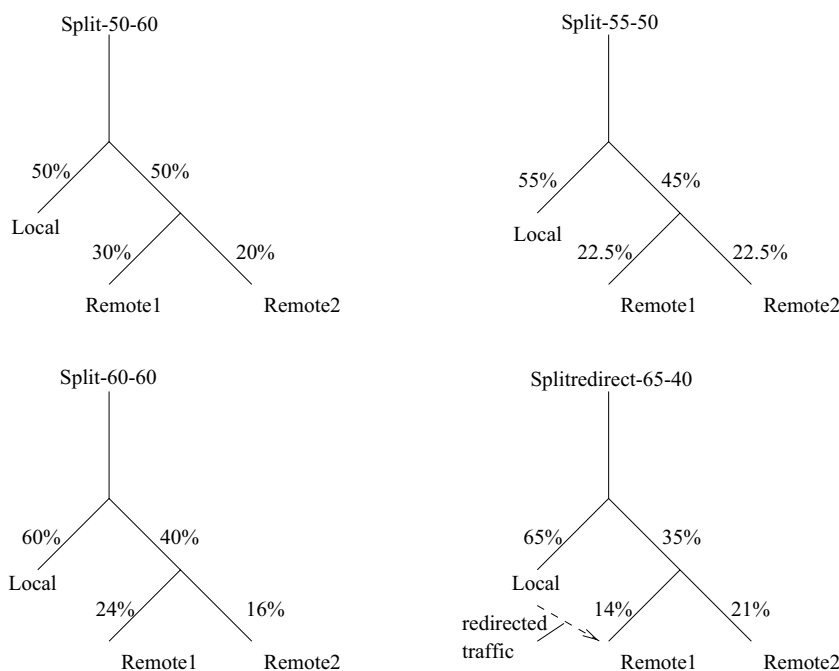


Fig. 5. Traffic to local and remote clusters for different splits.

objective is to compare split-based policies which divide the incoming traffic load in the same proportion as the resource capacities on the VoD system.

In Figs. 3 and 4, the performance of redirect policy is compared with various split-redirect and split policies. Blocking rates in number of requests blocked per hour are plotted as a function of varying load in requests per hour. Redirect policy has the minimum overall blocking rate in both cases. The redirect policy gives maximum utilization at each resource because remote service is required only when the local cluster cannot handle the load. Through these results, we show that even with only limited redirection it is possible to improve the performance of the VoD system.

Among split-based policies, the 55-50 split for the split policy and 65-40 split for the split-redirect policy have the lowest overall blocking rate. At low loads, the difference between split and split-redirect is minimal in which case an adaptive request handling policy could be devised by eliminating redirection.

*Difficulties With Split-Based Policies:* In the following paragraphs, we illustrate through several experimental results the difficulty in picking an efficient split for a given workload. Since the video collection is replicated on all clusters, finding the efficient split is that much easier. However, even in this case, we show through what we call divergence and crossover behavior of split-based policies, the difficulty in picking an efficient split for a given workload.

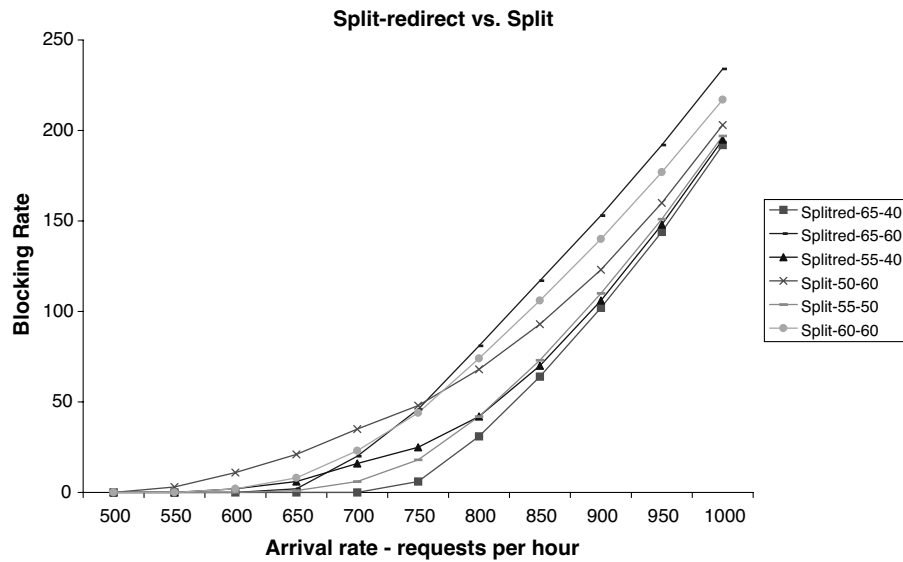


Fig. 6. Comparing split-redirect and split policies.

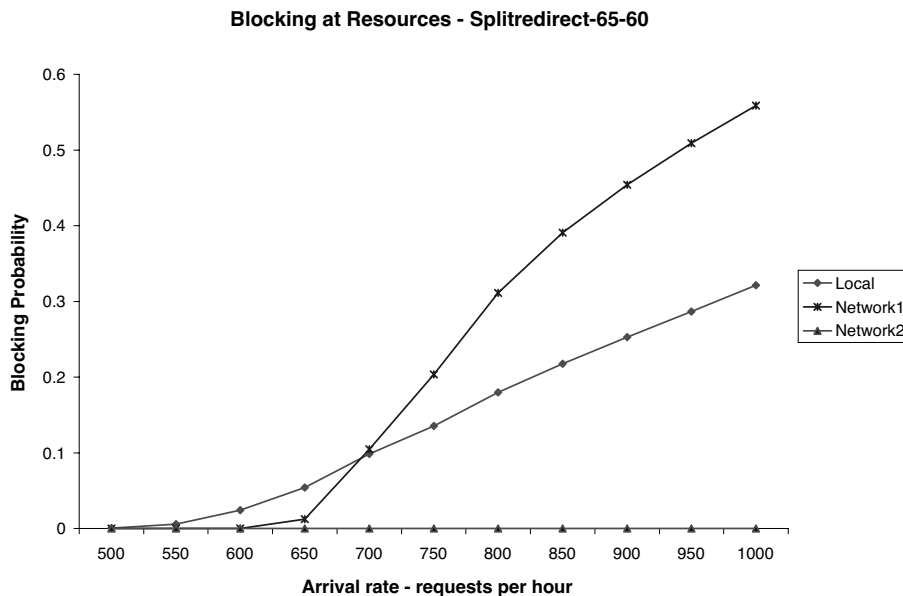


Fig. 7. Blocking at resources: split-redirect-65-60.

The difficulty of finding an appropriate split for a given load can be analyzed by explaining the interaction between various splits in the split-based policies which are characterized by *crossovers* and *divergences*. For instance, in Fig. 3, we see a crossover between split-50-60 and split-60-60 policies. At low loads of 500 to about 800 requests per hour, split-60-60 policy performs better than split-50-60 policy. To explain the reason for crossovers and divergences, we have to look at the actual percentage of traffic directed at each resource under a split-based policy (see Fig. 5). The actual traffic directed toward a resource will determine how efficiently that resource is being utilized. Under split-50-60 policy, the proportion of traffic going to local, remote cluster1 and remote cluster2 is 50, 30, and 20 percent respectively and under split-60-60 policy, it is 60, 24, and 16. The reason for the crossover is that at low loads, the local cluster is better utilized and networks are not congested yet with the split-60-60 policy. But at high

loads, both local and network1 are overutilized and network2 is under-utilized resulting in worse performance than split-50-60 policy. In Fig. 3, we notice a divergence behavior with the split-60-50 and split-60-60 policies. At low loads of up to 650 requests per hour, both policies have the same performance. The divergence occurs around 650 requests per hour with split-60-60 policy having worse performance. The reason for this is that at higher loads, network1 gets congested faster than the other resources for the split-60-60 policy. In general, crossovers are driven by the proportion of traffic directed toward the local cluster and divergence by the remote traffic. Also note that a certain split that performs well under low load conditions will deteriorate quickly as load is increased as shown for policy split-redirect-65-60 in Fig. 6.

Figs. 7–9, show blocking performance at individual resources for a particular split-based policy. Analyzing blocking at individual resources helps explain further the crossover

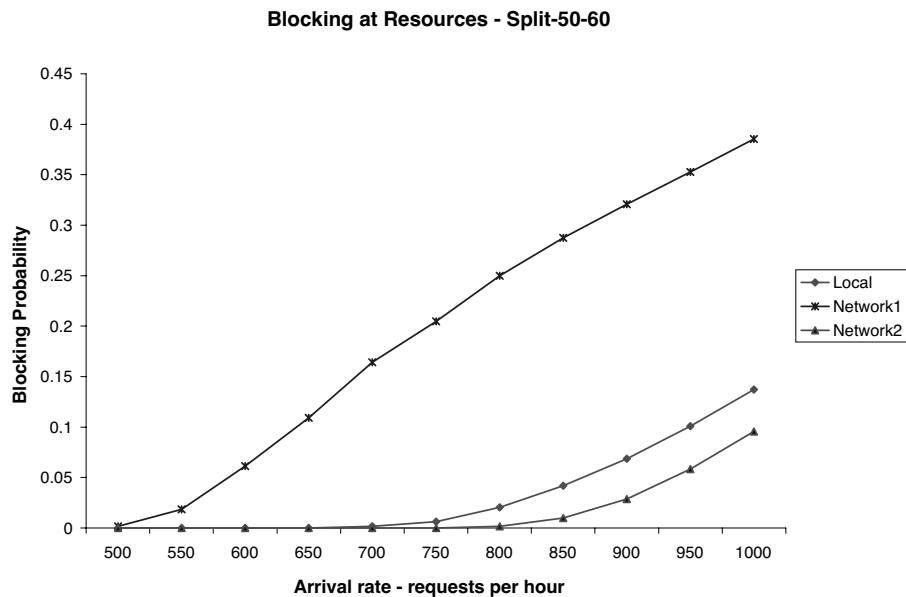


Fig. 8. Blocking at resources: split-50-60.

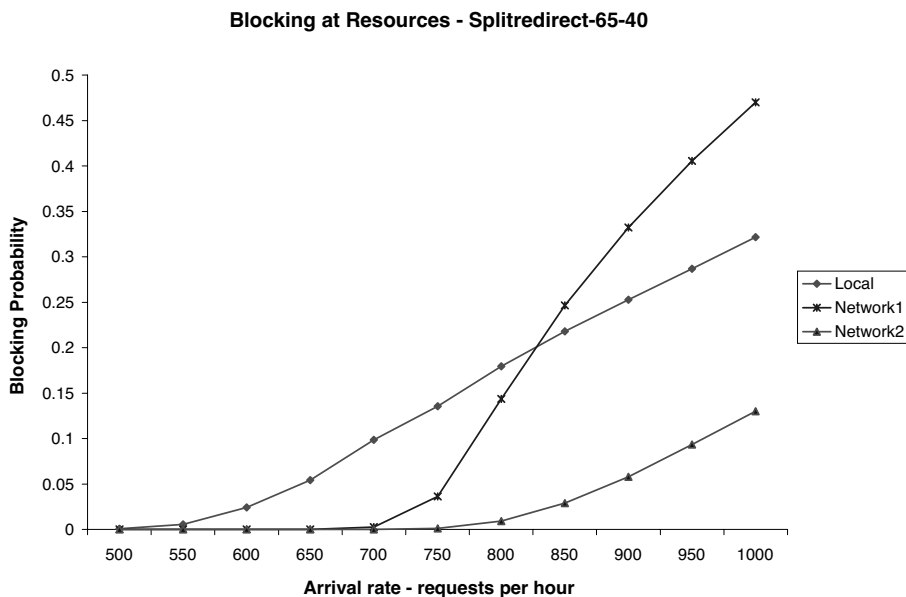


Fig. 9. Blocking at resources: split-redirect-65-40.

and divergence behavior; for example, the crossover between the split-redirect-65-60 policy and the split-50-60 policy. At low loads there is already a significant blocking activity at network1 for the split-50-60 policy (see Fig. 8), but blocking is more gradual. However, the crossover occurs because the performance of the split-redirect-65-60 policy deteriorates at a faster rate as a result of overloading network1 with redirected traffic from the local cluster and underutilization of network2 (see Fig. 7). A divergence between split-redirect-65-60 and split-redirect-65-40 is explained by the overloading of network1 at loads higher than 650 requests/h as shown in Fig. 7. The overloading of network1 is more gradual under the split-redirect-65-40 policy as shown in Fig. 9.

We can find efficient split policies by choosing splits that are close to the proportion of resource capacities within the hierarchical VoD system. In the VoD architecture, the stream han-

dling capacity at the local cluster is 750 requests and the network stream handling capacities are 311 requests each for the two routing paths to the remote clusters. Therefore, 55% of the overall VoD system capacity is at the local cluster with the networks providing 22.5% each. A precise split for the split-based policies is possible only if the portion of remote cluster capacity dedicated to each user population is known ahead of time. However, when remote clusters are dynamically shared with other user populations, it will be difficult to find efficient split policies and redirect policy will be the only alternative in that situation. Another factor that complicates finding an efficient split is the video distribution between local and remote clusters. The difficulty in finding accurate splits when only a partial video collection is available at the local cluster is discussed in Section IV-B. In Fig. 10, a comparison of policies with splits proportional to resource capacity available on local and remote clusters is

Policies Proportional to Resource Capacities

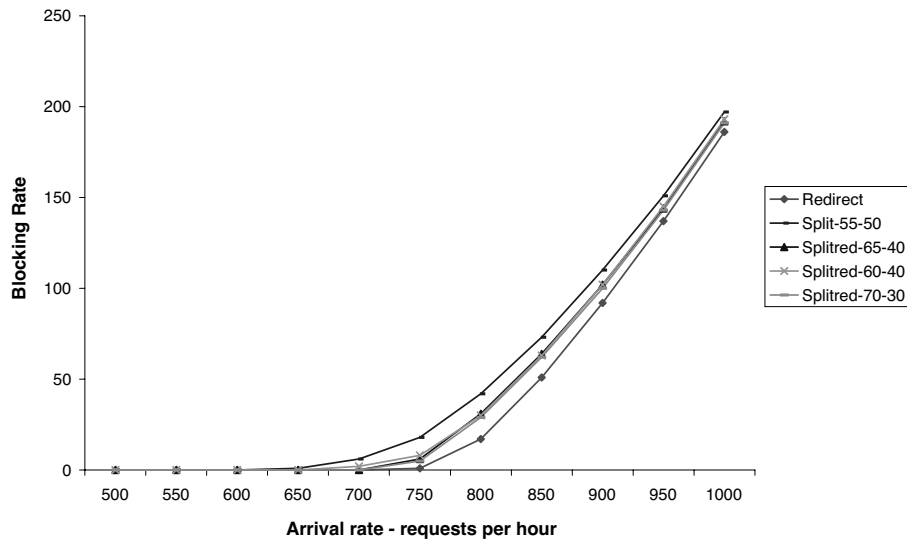


Fig. 10. Blocking rate: proportional splits.

TABLE III  
SCALABILITY DATA

Workload	Servers	Network1	Network2
170	1	498	498
340	2	996	996
510	3	1494	1494
680	4	1992	1992
850	5	2488	2488
1020	6	2988	2988
1190	7	3486	3486
1360	8	3984	3984
1530	9	4482	4482
1700	10	4980	4980

shown. Split-55-50 policy has the worst performance and redirect policy the best. Among split-redirect policies, 65-40, 60-40, and 70-30 splits have similar performance. Even a one-level redirection as provided by the split-redirect policies is enough to result in better performance than the most efficient split policy.

*Scalability:* In this set of experiments, the resource capacities in the VoD system are varied in proportion to the request workload to demonstrate that the overall blocking probability remains the same. However, the video collection size is kept constant for all experiments which will enable us to study the effect of video collection size over VoD system size. The experiment is constructed as follows—we fix the load per server and the network bandwidth on the two network paths and increase all four in lock step. For instance, at 170 requests per hour, the number of servers at the local cluster is equal to one and the network bandwidth is 498 Mbps. As we double the workload to 340 requests/h, we double the number of servers and the network bandwidth. Table III shows the values used in this experiment.

In Fig. 11, we show that the results of the proposed end-to-end analysis are scalable for the three request handling policies.

The overall blocking probability is plotted as a function of VoD system size. We show that the blocking performance levels off at about 5% for the redirect policy. The blocking performance for the split-redirect-65-40 policy is shown leveling off at 6% and for the split-55-50 policy it is shown to level off at 8%.

This result is important because it shows that as the VoD system size is scaled up from one server to ten servers at the local cluster with appropriate network bandwidth increases, the overall blocking stabilizes gradually and remains constant. One exception to this behavior occurs at one server capacity. The reason for high blocking for the one server capacity is that nearly 70% of the video collection has to be serviced from the remote clusters. Only the top 30 videos are stored in the local cluster because of limited storage capacity. Serving requests from the remote clusters overloads the network resulting in very high blocking. The other important factor contributing to the degraded performance is the lack of replication because we only have one server in the local cluster. The number of copies per video is limited to one. Therefore, the advantage of replication of top videos is lost leading to worse performance. Even at two-server capacity, blocking is still high for the same reason. As we increase the number of servers, the replication advantage becomes marginal.

The significance of replication in the VoD architecture is illustrated by cross referencing results from different experiments. The results shown in Fig. 12 indicate that it is better to have many smaller servers in the local cluster than one large server to take advantage of the replication scheme (see [5] for details on replication and data placement algorithms). The first scenario in Fig. 12 is part of the scalability results where we have a single server in the local cluster. The second scenario is part of the video distribution results from Section IV-B and has five servers in the local cluster. In both scenarios we maintain the same video distribution between local and remote clusters—top 30 videos are stored on the local cluster. However, given our replication algorithm, the top videos are replicated up to five times in the second scenario. Lack of replication

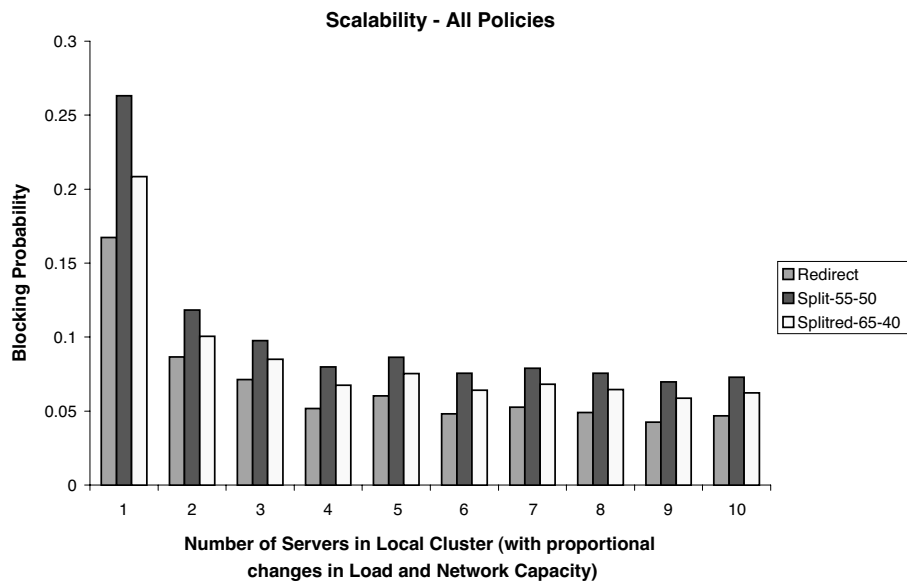


Fig. 11. Scalability: all policies.

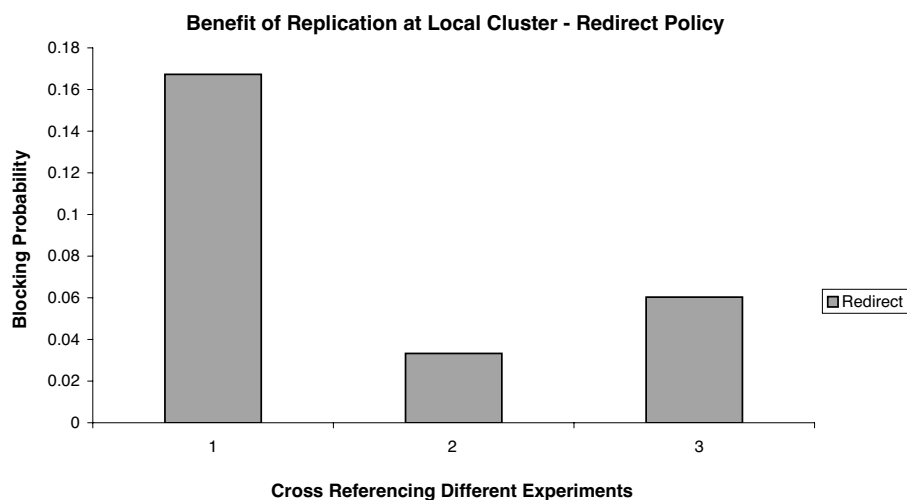


Fig. 12. Benefit of replication at the local cluster.

TABLE IV  
VIDEO DISTRIBUTION FOR DIFFERENT LEVELS OF STORAGE USED

Storage Used	Local	Remote
100	30	120
200	76	74
300	135	15
400	147	3
500	150	-

with just one server in the local cluster in the first scenario results in very high blocking performance at about 16%. The replication in the second scenario distributes the load on the five replicated copies of videos and reduces the redirected load on the networks resulting in low blocking probabilities. Using this result, we conclude that it is better to have many smaller

servers in a local cluster than one large server to take advantage of the replication process.

Scenario 3 is part of the scalability experiment where we have the complete video collection available at the local cluster. A comparison between scenarios 2 and 3 in Fig. 12 shows that distributed video collection in scenario 2 results in lowering overall blocking further. As discussed in Section IV-B, the partial allocation of videos to local cluster results in lower overall blocking probability because there is less contention of resources for the top videos at the local cluster. However, it also introduces higher blocking for videos serviced only from the remote clusters.

### B. Distributed Video Collection

The purpose of this experiment is to evaluate blocking performance of the VoD system when only a partial list of top videos are stored on the local cluster. Requests for the remaining videos will be served from the remote clusters. This partial allocation makes more stream capacity available to the top fraction of the

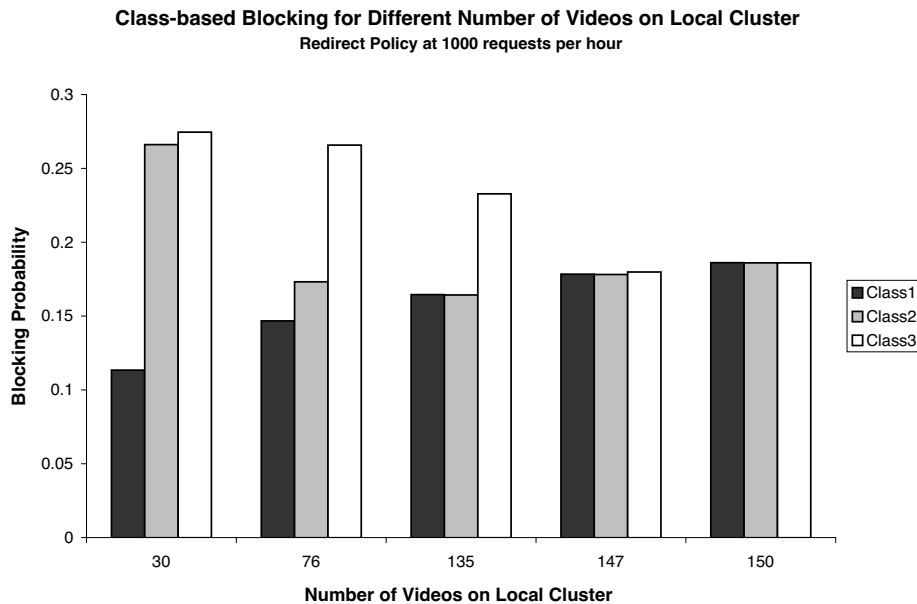


Fig. 13. Different video distribution: class-based blocking.

videos at the local cluster. This will reduce blocking and therefore, redirected traffic from the local cluster, but at the same time we expect increased blocking for videos served from the remote clusters only. We define three classes of videos based on popularity for this experiment: top 20% of the videos are class1 and the other 80% of the videos are divided into class2 and class3 videos. We analyze class-based blocking performance for several video distributions between local and remote clusters. The remote clusters are nonblocking in this experiment.

The basic idea behind the replication algorithm (see [5]) is to distribute the available storage capacity at the local cluster to videos in proportion to their popularity. Therefore, by varying the storage parameter in the replication algorithm we can vary the number of videos assigned to the local cluster. We vary the storage used on the local cluster so that the replication algorithm will allocate a partial to complete video collection to the local cluster. The *storage used* is varied from 100 to 500 GBytes per server in the local cluster. The local cluster size is still maintained at five servers. The video distribution between local and remote clusters at each level of storage used is given in Table IV. For instance, if used storage is limited to 200 GBytes per server on the local cluster, we only have the top 76 videos placed on the local cluster. The demand for the rest of the videos has to be exclusively met by the remote clusters. The stream handling capacity of the local cluster remains the same as before even as we vary the number of videos placed on the local cluster.

Partial allocation of the video collection to the local cluster results in different blocking performance for individual videos. If only the top 30 videos are stored on the local cluster, class 2 and class 3 videos which account for the bottom 80% of the video collection experience more than 25% blocking as shown in Fig. 13 at a load of 1000 requests per hour. At about top 75 videos on the local cluster, the bottom 40% of the videos still experience above 25% blocking. The decision to partially or completely allocate videos to local cluster must consider the fairness issue toward each video request. A replicated video collection

at all resources provides the same blocking probability for all videos whereas a distributed video collection will induce additional blocking for videos found on fewer resources because of reduced stream handling capacity available to those videos.

The request handling policy employed in this experiment is the redirect policy and the request load is fixed at 1000 requests/h. Split-based policies are not appropriate in a distributed video collection environment because of the difficulty in estimating the traffic split. It will require a priori computation about the video distribution according to the replication algorithm and storage capacities available. The appeal of split-based policies is their simplicity which will be lost if any computation is required ahead of time for finding an efficient split.

### C. Performance Summary for Redirect and Split-Based Policies

Split-based policies work well when the split in traffic mirrors the proportion of available capacity on all resources. In many instances, it is not easy to compute the available capacity especially if the resource is shared by many user populations. Efficient split-based policies are also difficult to find when the video collection is distributed between local and remote clusters. The split in traffic has to match the video distribution, the demand for the individual videos, and the available capacity of each resource and that requires prior knowledge and computation. Even in a replicated video collection scenario, our experiments showed that a particular split that works well for low workload quickly deteriorates for moderate and high workload. On the other hand, all of our experimental results show that the redirect policy, even with only limited redirection, has better performance than split-based policies. Redirect policies also provide simpler connection establishment semantics, although they have a higher implementation overhead as discussed in Section III-A. The tradeoff between the two classes of request handling policies is the computational overhead for finding efficient splits versus the extra connection setup time for the redirect policy.

The redirect policy is a better alternative to split-based policies under distributed video collection and nonblocking remote servers. Finding efficient splits requires a priori computation about the video distribution and the additional traffic to remote clusters which takes away the appeal of simple implementation for split-based policies. We showed that blocking for the top fraction of the most popular videos placed on the local cluster can be reduced with partial allocation of videos. However, videos served only from the remote clusters will experience higher blocking because of reduced resource availability. The overall blocking probability for distributed video collection is in general lower than that of replicated video collection scenario. The performance of the VoD system will diminish if remote clusters are shared with other user populations. Dynamically shared resources in the VoD architecture will generally further complicate determining efficient splits for split-based policies.

## V. CONCLUSION

In this paper, we developed a performance evaluation method for analyzing distributed VoD systems. We demonstrated the need for such an evaluation to determine efficient use of all resources in a VoD system with a hierarchy of servers and network elements. Using an extensive simulation of the distributed VoD architecture, we designed and evaluated several request handling policies. The significance of simulation results is that even limited redirection of blocked requests indicates superior performance. We showed that the redirect policy has simpler connection establishment semantics than split-based policies. We showed that finding efficient splits in a dynamically shared resource environment and for unknown video distribution is difficult. Incorporating redirection into signaling protocols will enhance performance.

## REFERENCES

- [1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Services*, Dec. 1998, IETF-RFC 2475.
- [2] E. Chang and A. Zakhor, "Scalable video data placement on parallel disk arrays," *Proc. SPIE*, vol. 2185, pp. 208–221, 1994.
- [3] L. Georgiadis, R. Guerin, V. Peris, and R. Rajan, "Efficient support of delay and rate guarantees in an internet," in *ACM SIGCOMM*, Aug. 1996, pp. 106–116.
- [4] K. Keeton and R. H. Katz, "Evaluating video layout strategies for a high performance storage server," *Multimedia Syst.*, vol. 3, no. 2, pp. 43–52, 1995.
- [5] P. V. Mundur, "An Integrated Approach to End-to-End Analysis of Distributed Video-on-Demand Systems," Ph.D. dissertation, George Mason Univ., Fairfax, VA, 2000.
- [6] A. Neogi and T.-C. Chiueh, "Performance analysis of an RSVP-capable router," *IEEE Network*, vol. 13, no. 5, pp. 56–63, Sept./Oct. 1999.
- [7] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control—the single node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344–357, 1993.
- [8] —, "A generalized processor sharing approach to flow control—the multiple node case," *IEEE/ACM Trans. Networking*, vol. 2, no. 2, pp. 137–150, 1994.

- [9] A. L. N. Reddy and J. Wyllie, "Disk scheduling in a multimedia I/O system," in *Proc. ACM Multimedia Conf.*, 1993, pp. 225–233.
- [10] P. S. Yu, M. S. Chen, and D. D. Kandlur, "Design and analysis of a grouped sweeping scheme for multimedia storage management," in *Proc. Third Int. Workshop on Network and Operating System Support for Digital Audio and Video*, 1992, pp. 38–49.
- [11] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: a new resource reservation protocol," *IEEE Network*, vol. 7, no. 5, pp. 8–18, Sep 1993.



**Padmavathi Mundur** (S'98–M'00) received the B.E. degree in industrial and production engineering from Bangalore University, India, the M.E. degree in systems engineering from the University of Virginia, Charlottesville, and the Ph.D. degree in information technology from George Mason University, Fairfax, VA, in 2000.

She is currently an Assistant Professor, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore. Her research interests include distributed systems, multimedia networking, and analytical performance modeling and resource allocation techniques.



**Robert Simon** received the B.A. degree in history and political science from the University of Rochester, Rochester, NY, in 1981, and the Ph.D. degree in computer science from the University of Pittsburgh, Pittsburgh, PA, in 1996.

He joined the Department of Computer Science, George Mason University, Fairfax, VA, in 1996, and is currently an Associate Professor. He previously worked at Citibank, The University of Pittsburgh Medical Center, and spent a summer at HP Labs, Palo Alto, CA. His research interests are in networks, real-time, and distributed systems. He has published over 50 peer-reviewed conference and journal papers in these areas.

Dr. Simon has served on numerous program committees and review panels, and was the Program Chair for the SCS Communication Networks and Distributed Systems Conference in 1999–2001.



**Arun K. Sood** (SM'83) received the B.Tech degree from the Indian Institute of Technology (IIT), Delhi, in 1966, and the M.S. and Ph.D. degrees from Carnegie Mellon University, Pittsburgh, PA, in 1967 and 1971, respectively, all in electrical engineering.

He is Professor and Chair in the Department of Computer Science at George Mason University, Fairfax, VA, and the Director of the Center for Image Analysis. He has held academic positions at Wayne State University, Detroit, MI, Louisiana State University, Baton Rouge, and IIT, Delhi. His research has been supported by the Office of Naval Research, National Imagery and Mapping Agency, National Science Foundation, U.S. Army Belvoir RD&E Center, U.S. Army TACOM, U.S. Department of Transportation, and private industry. He was awarded grants by NATO to organize and direct advance study institutes in relational database machine architecture and active perception and robot vision. His research interests are in image and multimedia computing, signal processing, parallel and distributed processing, performance modeling and evaluation, simulation and modeling, and optimization.