

Class-Based Access Control for Distributed Video-on-Demand Systems

Padmavathi Mundur, Arun K. Sood, and Robert Simon

Abstract—The focus of this paper is the analysis of threshold-based admission control policies for distributed video-on-demand (VoD) systems. Traditionally, admission control methods control access to a resource based on the resource capacity. We have extended that concept to include the *significance* of an arriving request to the VoD system by enforcing additional threshold restrictions in the admission control process on request classes deemed less significant. We present an analytical model for computing blocking performance of the VoD system under threshold-based admission control. Extending the same methodology to a distributed VoD architecture we show through simulation that the threshold performance conforms to the analytical model. We also show that threshold-based analysis can work in conjunction with other request handling policies and are useful for manipulating the VoD performance since we are able to distinguish between different request classes based on their merit. Enforcing threshold restrictions with the option of downgrading blocked requests in a multirate service environment results in improved performance at the same time providing different levels of quality of service (QoS). In fact, we show that the downgrade option combined with threshold restrictions is a powerful tool for manipulating an incoming request mix over which we have no control into a workload that the VoD system can handle.

Index Terms—Distributed video-on-demand (VoD) system, multirate service model, resource allocation, threshold-based admission control.

I. INTRODUCTION

THE RAPID advances in storage and compression technology along with the evolution of standards and protocols for high speed, integrated services networks has spurred interest in distributed multimedia systems. Of the many applications for distributed multimedia systems, video-on-demand (VoD) has much appeal because of its on-demand nature. Using a distributed VoD system, a client will be able to request a video from anywhere and at any time. In response to a client's request, VoD systems will deliver high quality digitized video directly to client set-top boxes or workstations. A typical VoD architecture consists of three subsystems: storage, network, and client. In such an architecture, admission control tests are employed on all subsystems before admitting a new request to check whether resources are available to provide guaranteed service. Traditionally, the concept of admission control takes into account only

the resource capacity. We extend that idea to include situations where a request may not be admitted into the system even if capacity is available so that the unused capacity is conserved for use by more lucrative requests.

The focus of this paper is to design admission control and request handling policies which take into account the inherent nature of an incoming request. For instance, a request for a more popular video might bring in more revenue and therefore, more such requests should be allowed into the system. In that case, we may want to conserve resources by rejecting requests for less popular videos in anticipation of popular video requests. We can accomplish that by limiting access to video requests for less popular videos by enforcing thresholds in the VoD admission control process. To induce a preference order among requests we introduce the idea of threshold-based admission control. In the first part of this paper, we develop an analytical model for computing blocking probabilities under threshold-based admission control in a single resource such as a video server. In the second part, we extend the threshold-based analysis to a distributed VoD architecture using simulation. The threshold-based analytical model at the video server gives us a quick and easy way to accurately assess different threshold-based policies. However, there are limitations to an analytical approach when we consider extending it to the distributed architecture and the large scale of a VoD system.

The following major factors complicate the development of an analytical model of the distributed VoD architecture: 1) multiple resources on multiple hierarchies and a large video collection; 2) large number of videos and request classes, each differing in its relative popularity and significance; 3) blocking and not queueing requests for service; 4) redirection of blocked requests to other parts of the VoD system; and 5) multiple classes of requests with one class being transformed into another for service after being blocked. For instance, a high bitrate request gets downgraded to a low bitrate request. Reflecting all these features in a blocked queueing system of the Erlang loss type [1] is difficult and may even be impossible, necessitating a simulation study to model the full complexity of a distributed VoD architecture. The goal of this paper is to address some of these limitations by using an extensive simulation. Toward that end, the main contributions of this paper are as follows.

- 1) We show that it is advantageous to distinguish between request classes in the admission control process in a VoD system based on their merit. We validate the simulation results through analytical modeling and show that the VoD performance behavior is the same under threshold-based admission control.

Manuscript received November 4, 2002; revised March 24, 2003. This paper was recommended by Associate Editor H. Watanabe.

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).

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

Digital Object Identifier 10.1109/TCSVT.2005.848351

- 2) Through analytical means, we show that we can compute blocking probabilities under threshold-based admission control efficiently.
- 3) Through simulation, we implement the threshold-based admission control process in the complex VoD architecture consisting of server hierarchies and network paths. We use multiple classes for requests with different playback rates and a large video collection with differing popularity of videos.
- 4) We explicitly model the network for providing guaranteed services essential for multimedia data transfer. We use Weighted Fair Queueing (WFQ) [2] scheduling to derive reserved rates at the network.
- 5) We propose several request handling policies, the most significant of which is the policy that allows downgrading a blocked request to lower bitrate service. We show that the downgrade option combined with threshold restrictions results in improved performance and at the same time acts as a powerful tool for manipulating an incoming request mix over which we have no control into a workload that the VoD system can handle.

This paper is organized as follows. In Section II, we discuss related work. In Section III, we present the distributed VoD architecture and define admission control conditions at the server and the network. In Section IV, we develop a computationally efficient analytical model to compute blocking probability under threshold-based admission control. A comprehensive simulation of the distributed VoD architecture is presented in Section V along with significant results. We conclude the paper in Section VI.

II. RELATED WORK

Admission control algorithms at a video server determine whether a new client can be admitted without violating the continuous playback requirements of the clients already being served; that is, the time needed to retrieve the blocks for existing clients should not exceed the minimum playback duration of the blocks retrieved for an existing client. Two classes of admission control strategies are discussed in other researches: deterministic and statistical [3]–[7]. All these and other related works focus on enforcing capacity-based admission control in the disk subsystem and do not consider different requests classes or the distributed VoD architecture.

In [8], Chen introduces the idea of distinguishing between high and low priority classes, but the resource capacity is partitioned between the two classes. Admission control is enforced on the dedicated partition for each class. The loss rate is found by modeling each dedicated partition of server capacity as a separate $M/M/C/C$ queue. In this paper, the admission control scheme discussed in [8] is called *complete partitioning* admission control. Aein refers to it as dedicated access in [9]. Blocking in dedicated access is easily computed using Erlang loss model whereas computing blocking for *complete sharing* or *partial sharing* based on thresholds requires computational algorithms. Distinguishing between request classes for providing access based on their merit is advantageous. Some of the research in resource allocation algorithms [9], [10] mention this

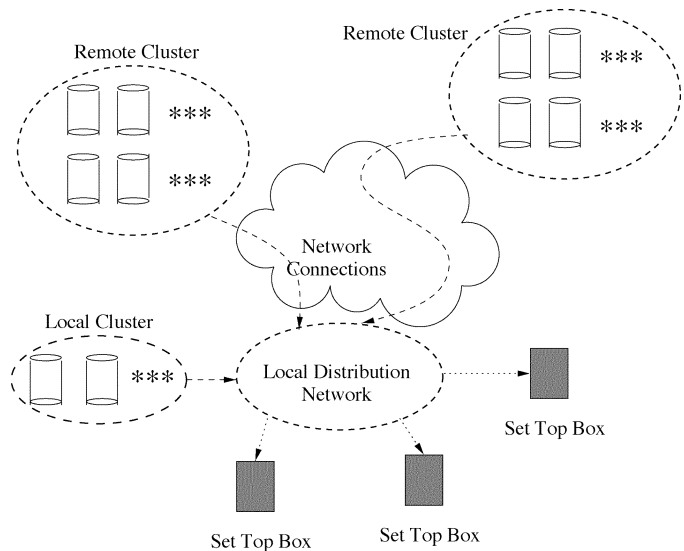


Fig. 1. Hierarchical VoD architecture.

advantage in the context of a single resource but computation of blocking probabilities for threshold-based resource usage has not been done before. More recently, Chan and Tobagi [11] discuss the profitability of batch arrivals to a central video server and enforcing threshold restriction on some arrivals to maximize profits.

This work is different from other works in VoD research for the following reasons. We extend the concept of capacity-based admission control to include additional attributes of request/video classes in the admission control process. We present both analytical and simulation models and show where former fails, we can use latter to analyze the complex VoD architecture. We also explicitly model the network in the simulation using fair queueing algorithms to provide service guarantees. We experiment with several request handling policies that include threshold restrictions and evaluate them using global metrics.

III. DISTRIBUTED 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 many user populations. The hierarchical VoD system architecture used in this analysis 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. The set-top boxes at the client site provide the 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 referred to as the *reference* user population is shown in Fig. 1.

Each local cluster is dedicated to its reference user population. The local cluster may store a complete or a partial set of videos from the video collection. The service from the local cluster is provided over a local distribution network, such as an ATM LAN or xDSL. The requests originating from a reference

user population are best served by the 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.

A capacity-based 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 and n_c is the number of requests currently in service. A round is defined as the playback duration of the data sent to the client periodically. The request will be admitted into the system if the following condition holds:

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

Remote servers also provide video delivery service over high speed networks. The network context in our proposed model is a packet switching, IP-based network. The underlying scheduling scheme at the network is required to provide bounded delays with respect to a minimum reserved rate at the network. In the simulation, *weighted fair queueing* (WFQ) scheduling scheme, a well understood and most widely used scheme is assumed to be implemented in the network. In [2], 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 network model based on WFQ used in the simulation is briefly described below (see [12] for more details).

Using the WFQ model, the required minimum rate, R_r is derived corresponding to an input delay bound D_{\max} , in the following way. This required minimum rate which is at least as large as the playback rate is used in the admission control test before admitting a new request on the network

$$R_r = \frac{B_r + (J - 1)M}{D_{\max} - \sum_{j=1}^J \left(\frac{M_{\max}}{A_j} \right)} \quad (2)$$

where M_{\max} is the maximum size of a packet permitted in the network (MTU), A_j is the overall bandwidth on j th link or hop on a routing path with J links, B_r is the size of the data sent per round to the receiver, M is the maximum packet size for the flow.

Admission control tests for new flow requests are run on a link by link basis over the routing path. If enough resources are available, they are reserved for the new flow using a signaling protocol, such as RSVP [13]. 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 (2). If the *new* flow requests a rate of $R_{r_{\text{new}}}$, it will be admitted only if the following relation holds:

$$\sum_{i=1}^T R_{r_i} + R_{r_{\text{new}}} \leq A_j. \quad (3)$$

IV. THRESHOLD-BASED ADMISSION CONTROL AT A VIDEO SERVER

In this section, we develop an analytical model for evaluating the performance of a single video server under threshold-based admission control. An admission control *policy* determines whether an arriving request for a certain class of video should be admitted given the current state of the VoD server. Three types of class-based admission control are defined: *complete partitioning*, *complete sharing*, and *threshold type*. A complete partitioning policy means that the server capacity is completely partitioned and every class of request has access only to a dedicated set of resources. A complete sharing policy is one where requests of all classes are always accepted if there is available capacity. A threshold type policy allows requests of classes up to the thresholds assigned to those classes. A threshold *class-2* policy limits class 2 requests up to a specified threshold whereas always accepting class 1 requests if there is available capacity. Threshold type policies are useful because it may not be desirable to share all of the server capacity equally among classes. For instance, requests for certain classes of videos might bring in more revenue.

For simplicity, we explain our model using two classes. Without loss of generality, suppose there are two classes of requests and the vector (n_1, n_2) describes the state of the system in terms of the number of class 1 requests, n_1 and the number of class 2 requests, n_2 . The state of the VoD server is describable as a state space constrained by capacity. A capacity constraint is defined by $(\sum_{k=1}^2 n_k \leq C)$ where C is the maximum number of simultaneous requests that a video server can handle. The state space can further be constrained by other restrictions, such as when the admission control policy favors one class more than the other by enforcing a threshold restriction on one of the classes.

Given a state space of the system based on the admission control policy, we can identify those states where an arrival will be blocked either because of the capacity constraint or the threshold. Blocking probability is computed using this information. For example, Fig. 2 illustrates the situation for two classes of requests and maximum capacity of five requests at the video server. The policy illustrated in the figure is to allocate no more than 60% of the resources at the video server to class 2 requests. Therefore, only a maximum of three class 2 requests are allowed in the system. There are no restrictions on class 1 requests other than the capacity of the system. Because of capacity and policy restrictions, arrivals of the pertinent classes are rejected when the VoD system is in one of the blocking states.

The state space, Ω , for the example in Fig. 2 is defined by $\Omega = \{(n_1, n_2) : 0 \leq n_1 \leq 5 \text{ and } 0 \leq n_2 \leq 3 \text{ and } \sum_{k=1}^2 n_k \leq 5\}$

Blocking states for class 1 requests, $\Omega_1 = \{(2, 3)(3, 2)(4, 1)(5, 0)\}$ and blocking states for class 2 requests, $\Omega_2 = \{(0, 3)(1, 3)(2, 3)(3, 2)(4, 1)(5, 0)\}$.

A. Analytical Model

Two classes of videos, *popular* and *unpopular* are assumed to be stored on the video server. Each class consists of many videos of different duration. The duration of the video playback

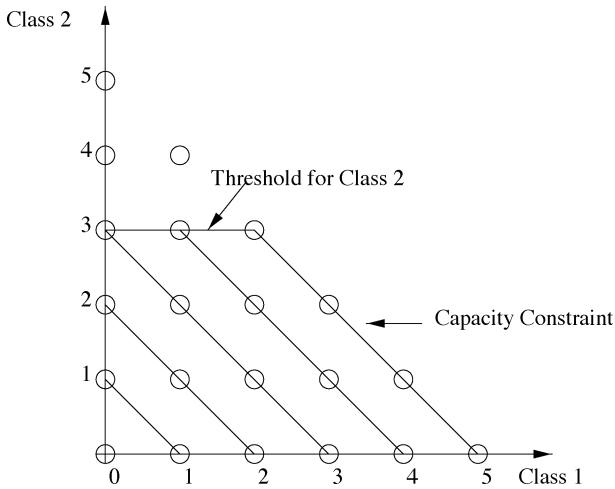


Fig. 2. Planar states for distribution q_i .

TABLE I
MODEL PARAMETERS FOR THRESHOLD ANALYSIS

λ_1	Arrival rate for popular movies
λ_2	Arrival rate for unpopular movies
$1/\mu_1$	Service time for popular movies
$1/\mu_2$	Service time for unpopular movies
C	Max. num. of requests at video server
l_k	Threshold for class k , $k = 1, 2$
P_{b_1}	Blocking probability for class 1 requests
P_{b_2}	Blocking probability for class 2 requests
$P(n_1, n_2)$	Probability of n_1 class-1 requests and n_2 class-2 requests, $\sum_{k=1}^2 n_k \leq C$

defines the service time for a video request. The service time is assumed to be exponentially distributed. The steady state distribution provided in this section, however, holds for any arbitrary service time distributions [14]. The arrival rates of requests per class are based on Poisson process. All incoming requests are blocked if the video server capacity is not available or if there are requests up to the threshold for that class. The admission control policy used is threshold type. Blocking probability for class 1 (popular) and class 2 (unpopular) requests are now computed using the approach below. Model parameters and definitions are shown in Table I.

As shown in [9] and [14], the equilibrium probability, $P(n_1, n_2)$, of finding the video server in the state (n_1, n_2) has the following product form:

$$P(n_1, n_2) = \frac{1}{G} \prod_{k=1}^2 \frac{\rho_k^{n_k}}{n_k!}, \quad \text{all } (n_1, n_2) \in \Omega \quad (4)$$

where the normalization constant is given by

$$G = \sum_{(n_1, n_2) \in \Omega} \prod_{k=1}^2 \frac{\rho_k^{n_k}}{n_k!}$$

and the traffic intensity is given by

$$\rho_k = \frac{\lambda_k}{\mu_k}$$

The blocking probability for the k th class P_{b_k} is given by

$$P_{b_k} = \sum_{(n_1, n_2) \in \Omega_k} P(n_1, n_2) = \frac{G(\Omega_k)}{G} \quad (5)$$

where $\Omega_k = \{(n_1, n_2) | (n_1, n_2) \in \Omega, (n_1, n_2) + \bar{e}_k \notin \Omega\}$ where \bar{e}_k is a vector equal to either (0,1) or (1,0), where the 1 is in the k th position. Ω_k represents the set of blocking states and $G(\Omega_k)$ is the sum of unnormalized probability of blocking states in Ω_k . Ω_1 and Ω_2 are the set of blocking states for class 1 and class 2 requests respectively.

The state enumeration technique discussed above for computing blocking probability is computationally very expensive. The need for computational efficiency arises because the cardinality of Ω grows roughly as C^k . Hardware limitations rule out computing G for even moderate sized problems. Some previous studies in finding computationally tractable solutions for computing G were done in the context of multirate circuit switched networks [14], [15] and multiprogramming [16]. For instance, Kaufman [14] and Roberts [15] independently developed recursive relations to compute the G factor. All these recursive solutions including Buzen's in [16] refer to the complete sharing policies where requests of all classes are always accepted if there is available capacity. In contrast, under threshold-based admission control, requests of certain classes are not admitted into the system beyond a threshold even if there is available capacity. We present a modified recursion-based result for threshold-based sharing policies in the following paragraphs.

1) *Blocking Probability Using Recursion:* We develop an efficient way of computing blocking probabilities for a single resource, multiserver, two-class threshold type system here. Our approach follows the Kaufman/Roberts recursion [14], [15] modified to compute blocking probability for a multirate tree network in [17]. In a multirate tree network, both access link and common link capacities have to be available for call establishment. The proposed threshold-based admission control situation can be analyzed based on the same logic. For instance, a video request with threshold restriction can only be admitted if there is capacity and if the number of requests currently being served is less than the threshold. Therefore, threshold type policies introduce additional blocking states that should be included in the computation of blocking probabilities. Also, the state space Ω is now truncated according to the threshold values. A new function has to be included in the Kaufman–Roberts recursion to reflect additional blocking from the threshold blocking states.

The basic idea behind the recursion is summing $P(n_1, n_2)$ over Ω recursively along disjoint parallel planes on the diagonal, as shown in Fig. 2. From this summing technique, we get a distribution in terms of the total number of requests being served. Denote the total number of requests served as $I = \sum_{k=1}^2 n_k$, and define a state space

$$\Theta(i) = \left\{ (n_1, n_2) \in \Omega : \sum_{k=1}^2 n_k = i \right\}.$$

The distribution q_i , indicates the total number of requests being served at the video server from all classes

$$q_i = P(I = i) = \sum_{(n_1, n_2) \in \Theta(i)} P(n_1, n_2) \text{ for } i = 0, 1, \dots, C. \quad (6)$$

Define $G(i)$ the unnormalized probability of finding i requests in the VoD server, as

$$G(i) = \begin{cases} \sum_{(n_1, n_2) \in \Theta(i)} \prod_{k=1}^2 \frac{\rho_k^{n_k}}{n_k!}, & i = 0, 1, \dots, C \\ 0, & \text{otherwise.} \end{cases}$$

Multiplying (6) with G and substituting for $P(n_1, n_2)$ from (4) we get the expression above for $G(i)$

$$G(i) = q_i G \quad (7)$$

To handle threshold type policies, we define a threshold value which is less than C to limit the number of requests for those classes. Define $l_k < C$ as the threshold for class k requests. Define for $k = 1, 2$

$$B_k(i) = \begin{cases} P(I = i, n_k = l_k)G, & \text{if } l_k \leq i \leq C - 1 \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

$B_k(i)$ is the unnormalized probability of being in one of the additional blocking states due to the threshold. If no threshold is defined for a particular class, $B_k(i)$ for that class is 0. Blocking probability for k th class under threshold-based restrictions is given by

$$P_{b_k} = P(I = C) + P(I \leq C - 1, n_k = l_k) \\ P_{b_k} = \frac{G(C) + \sum_{i=l_k}^{C-1} B_k(i)}{\sum_{i=0}^C G(i)}. \quad (9)$$

To compute blocking probability, we need only calculate the G factor and $B_k(i)$, $i = l_k, \dots, C - 1$, for $k = 1, 2$. We use the following recursion to accomplish that.

Theorem 1:

$$G(i) = \frac{1}{i} \sum_{k=1}^2 \rho_k [G(i-1) - B_k(i-1)], \\ \text{for } i = 1, \dots, C \text{ and } G(0) = 1$$

The proof of Theorem 1 is contained in the Appendix.

Computing $B_k(i)$ for a threshold class-2 system is presented next. A threshold class-2 system allows class 2 requests up to a threshold and always accepts class 1 requests if capacity is available. In the analytical model, it is reasonable to limit class 2 (unpopular) requests and allow more of class 1 (popular) requests to be accepted into the system. Therefore, a threshold restriction is not assumed for class 1.

Similar to [17], a function $r(i)$ is defined to compute $B_2(i)$ using the same recursion from Theorem 1 but on class 1 dimension. The function $r(i)$ represents the resource usage from class

1 in excess of the class 2 threshold. Notice that the resource usage for class 2 is already at its threshold l_2

$$r(i) = \frac{1}{i} \rho_1 [r(i-1) - B_1(i-1)] \\ \text{for } i = 0, \dots, C - 1 - l_2 \text{ and } r(0) = 1 \quad (10)$$

Since we do not have a threshold restriction for class 1, by definition we have, $B_1(\cdot) = 0$. The equation above for $r(i)$ becomes

$$r(i) = \frac{1}{i} \rho_1 [r(i-1)].$$

We use the function $r(i)$ to compute $B_2(i)$ as follows:

$$B_2(i) = \left(\frac{\rho_2^{l_2}}{l_2!} \right) r(i - l_2), \quad i = l_2, \dots, C - 1. \quad (11)$$

Stripping off the recursion in (11) we have

$$B_2(i) = \left(\frac{\rho_2^{l_2}}{l_2!} \right) \left(\frac{\rho_1^{(i-l_2)}}{(i-l_2)!} \right)$$

which can be proved from equilibrium conditions, the product form expression (4), and the definition of $B_2(i)$.

2) *Remarks:* The threshold-based analysis need not be limited to a two class model. The model can easily be extended to K classes. The recursion in Theorem 1 is straight forward enough in that the summation will include K classes. The analysis gets complicated in the computation of $B_k(i)$ s. The presence of interference between threshold planes requires further recursion in $r(i)$ when computing a certain $B_j(i)$, $j \neq k$. If there is no interference, the computation of $r(i)$ is as shown below because all $B_k(\cdot)$ s will be zero.

$$r(i) = \frac{1}{i} \sum_{k=1, k \neq j}^K \rho_k [r(i-1)], \quad i = 1, \dots, C - 1 - l_j. \quad (12)$$

B. Numerical Results

In this section, we demonstrate how we can rapidly assess different threshold policies for a VoD server with two classes of requests, class 1 (popular) and class 2 (unpopular), and an overall capacity of $C = 500$ using the results from the previous section. No threshold restrictions are assumed on class 1 except the capacity of the server; but there is a threshold restriction on class 2. The recursion $r(i) = (1/i)\rho_1[r(i-1)]$ is used to compute all $r(i)$ s and (11) is used to compute $B_2(i)$. Once all $B_2(i)$ s are determined, Theorem I is used to compute $G(i)$.

Using (9), blocking probability for both classes is then calculated as follows:

$$P_{b_1} = \frac{G(C)}{\sum_{i=0}^C G(i)} \quad (13)$$

$$P_{b_2} = \frac{G(C) + \sum_{i=l_2}^{C-1} B_2(i)}{\sum_{i=0}^C G(i)}. \quad (14)$$

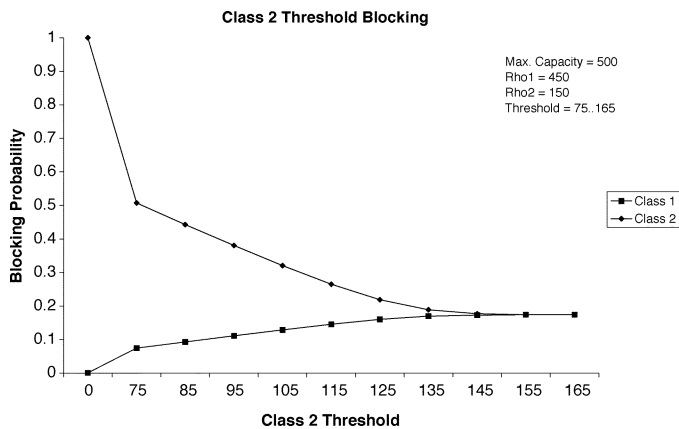


Fig. 3. Class 2 threshold versus blocking probability.

A computer program was written to compute blocking probabilities using (13) and (14). Fig. 3, shows the effect of varying the admission threshold for class 2 while ρ_1 and ρ_2 are held constant. In this analysis the capacity C is set to 500 (simultaneous requests), the threshold for class 2 ranges from 0 to 165, ρ_1 is set to 450 and ρ_2 is set to 150. As the threshold for class 2 is increased allowing more class 2 requests into the system, the blocking probability for class 1 increases as a result of increased sharing of resources. Eventually, we have a situation similar to complete sharing policy, where each class has the same blocking probability.

V. VoD SIMULATION AND RESULTS

In this and the following sections, we extend the threshold based analysis to a distributed VoD system using a simulation and show that the simulation results conform to the analytical behavior under threshold restrictions. The simulation was conducted using the CSIM simulation package [18].

A. Global Performance Metrics

In the VoD system, a request that gets rejected by the end-to-end admission control test is said to be *blocked*. The number of blocked requests forms the basis for developing global performance metrics such as overall blocking probability for the entire VoD system. Blocking probability per class is defined as the ratio of the total number of blocked requests in that class to the total number of arrivals at the VoD system for that class.

The performance metric for threshold-based analysis is the class-based blocking probability. In generating class-based performance analysis, we distinguish between single-rate and multirate service. For single-rate service, as the name implies all requests for any video in the video collection are served at a known, single rate. We define request classes for a single-rate service based on video popularity. To simplify the analysis for a large video collection, we divide the collection into a few classes based on video popularity. For instance, requests for the top 20% of the video collection can be thought of as a distinct class from the bottom 20% or the middle 60%.

For a multirate service, threshold-based analysis is conducted by defining classes based on a set of playback rates that the service provider offers. For instance, in a given user population,

we can estimate 60% of the users requesting 16 Mb/s playback rate, another 20% requesting 8 Mb/s rate and the remaining 20% requesting a 4-Mb/s rate. Given this request mix, an arriving request for any video has 0.6 probability of requesting a 16 Mb/s playback rate. Threshold-based analysis using class-based blocking is conducted based on classifying an arriving request on the requested playback rate. The class-based blocking probability for the j th class of requests is defined as the ratio of the total number of blocked requests to the total number of arrivals for all videos in the video collection from j th class of requests at the VoD system.

The overall blocking probability for the VoD system is defined as follows. The request arrival rate at the VoD system is λ . Let the probability of a request for i th video be p_i which for instance, can be determined using Zipf distribution [19]. The arrival rate for i th video is $\lambda_i = \lambda p_i$. Let m_j be the probability that a request is from the j th class which can be determined from the given request mix of the user population. The blocking probability of a request, b_{ij} for the i th video from j th class is defined as the ratio of the total number of blocked requests to the total number of arrivals for that video from that class at the VoD system. 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. The overall blocking probability at the VoD system, P_{vod} is the weighted average over all videos V and classes R , of the blocking probability of a request for i th video from j th class

$$P_{\text{vod}} = \sum_j^R \sum_i^V m_j p_i b_{ij}. \quad (15)$$

B. Simulation Setup

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 (see Fig. 1). 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. The request arrival process is assumed to be a Poisson process.

Each local cluster server is modeled after a 500 GBytes RAID-style server that can simultaneously serve 150 request streams of 8 Mb/s. The remote clusters are archival in nature and store all videos. 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 video collection consists of 150 videos and the demand for each video depends on the popularity of the video. Zipf distribution is used to find the probability of choosing k th video from a collection of V videos.

The following request handling policies are defined for evaluating the performance in this simulation. Evaluating VoD performance in terms of these policies is a way of quantifying resource usage and the resulting blocking performance. The significance of such an approach in our simulation is the evaluation of several alternative operational conditions under a unified framework.

TABLE II
SIMULATION DATA

Servers in local cluster	5
Storage capacity per server	500 GB
Disk transfer rate at server	1.2 Gbps
Hops to remote cluster1	3
Hops to remote cluster2	6
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 GB	2.46 to 4.8
Service time in hrs	0.68 to 2.03

- 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 and Split—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. In addition, all blocked requests from local cluster are redirected to remote cluster1. Under split-based policies blocked requests at local cluster are not redirected to remote clusters.
- Redirect with downgrade—Blocked requests under redirect policy are downgraded to the lowest bitrate where possible.

C. Threshold-Based Performance Evaluation

In this section, we present results of performance evaluation from three experiments.

- In the first experiment, we serve all requests at the same playback rate of 8 Mb/s leading to single rate service. The request classes are defined based on the popularity of videos that the requests are for. For instance, a client request can be for a video in the top 20% popularity and will be served at a standard playback rate of 8 Mb/s.
- In the second experiment, requests are served at different playback rates, each rate defining a class of requests leading to multirate service. Video popularity still determines the demand for videos. But the client is able to specify a playback rate for the video request.
- The third experiment is in the context of multirate service where we introduce the option of downgrading a blocked request to a lower bitrate service with and without threshold restrictions.

1) *Single Rate Service*: The goal of this experiment is to show that the performance results from the simulation are similar to the analytical results from Section IV-A. Request classes are defined based on the popularity of videos. Since we have an ordered list of videos based on popularity, a class 1 request is defined as a request for any of the top 20% of the videos; a class 2 request as a request for any of the middle 40% of the videos; a class 3 request for the bottom 40%. The threshold values for

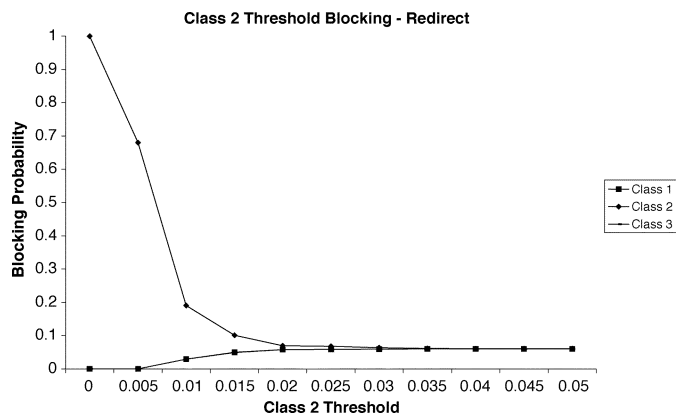


Fig. 4. Class 2 threshold versus blocking probability: redirect policy.

class 2 and class 3 are in the range of 0%–5% of the capacity. A 0% threshold restriction for class 2 means that all class 2 requests will be blocked.

In Fig. 4, the effect of threshold blocking under redirect policy is shown. Class-based blocking is evaluated as a function of threshold restrictions on either class 2 or class 3 requests. The behavior shown matches that of the analytical model from Section IV-A. The threshold experiments are conducted at a request load of 850 requests per hour. Requests for class 2 and class 3 consisting of the less popular videos are subjected to threshold restriction. In Fig. 4, class 2 threshold restriction results in 0% to 5% blocking for class 1 and class 3 videos. Blocking for class 2 is at 100% at 0 threshold, meaning that no requests of class 2 are accepted into the system. Blocking levels off around 5% for all three classes at about 3% threshold when the threshold has no effect since the arrival rate is below the threshold anyway. Performance impact is similar when class 3 threshold restrictions are enforced. Similar analysis is applied to split and split-redirect policies and the behavior shown conforms to the analytical model from Section IV-A (see [20]).

The relative performance order among the three policies is that the redirect policy has the best performance with the split-based policies having the worst blocking performance (see [12], [20] for further details). The threshold-blocking performance for the redirect policy stabilizes well before either of the split or split-redirect policies. The implication of this is that for the same threshold value, blocking for all requests will be higher for split-based policies.

2) *Multirate Service*: The classes for this experiment are defined based on the requested playback rate of an incoming request. We assume three classes based on the playback rates: requests are served at three different playback rates—4, 8, and 16 Mb/s. A request at 4 Mb/s playback rate is regarded as a class 1 request, at 8 Mb/s as a class 2 request, and at 16 Mb/s as a class 3 request. We estimate that x percent of the user population requests 4 Mb/s and y percent requests 8 Mb/s and the remaining, z percent requests 16 Mb/s giving us a certain (x, y, z) request mix. The request handling policy used in these experiments is the redirect policy since it exhibits the best performance among request handling policies.

In Figs. 5 and 6, we show blocking under class 3 threshold restrictions for request mixes (20,20,60) and (20,40,40) for a load

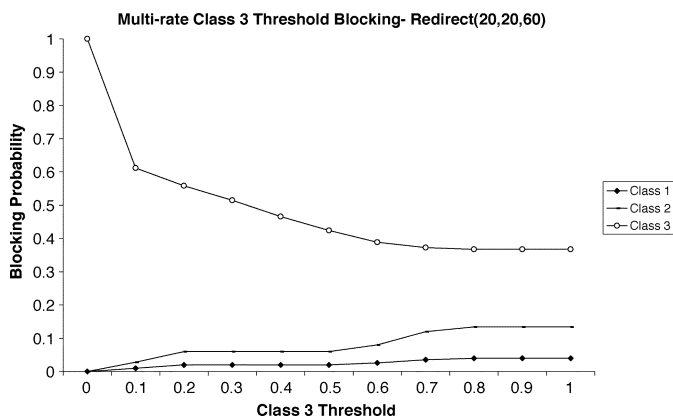


Fig. 5. Multirate results for class 3 thresholds at (20,20,60) request mix.

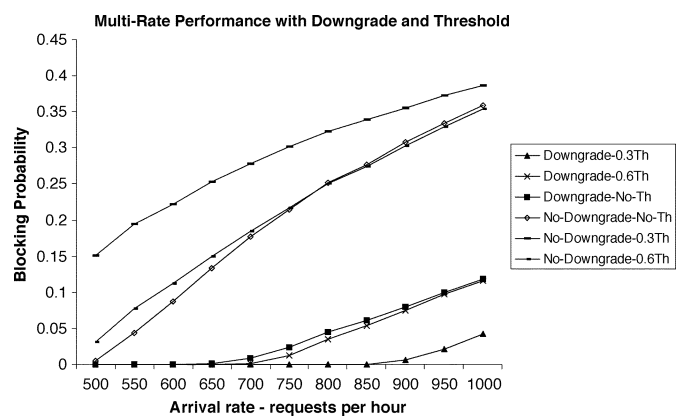


Fig. 7. Multirate results with downgrade option, (20,20,60) request mix.

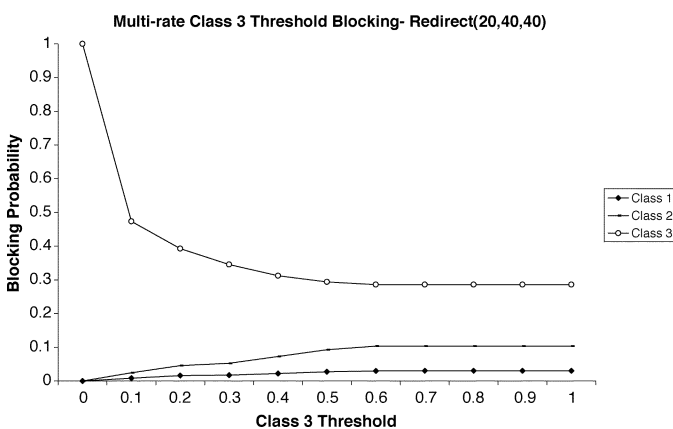


Fig. 6. Multirate results for class 3 thresholds at (20,40,40) request mix.

of 800 requests per hour. At 0 class 3 threshold, blocking probability for class 3 is equal to one because no class 3 requests are admitted into the system. As we relax the threshold restriction on class 3 requests, we observe that class 3 blocking decreases but blocking for other two classes increases. Threshold restrictions are applied to only class 3 requests since they have the highest resource requirement.

The performance behavior is the same as the analytical model from Section IV-A but with one key difference. Because of the different resource requirements of these classes, we see that each class levels off at different blocking performances as shown in Figs. 5 and 6 at class 3 threshold 1. We notice that class 3 requests at 16 Mb/s rate experience more blocking than class 1 requests at 4 Mb/s.

Imposing thresholds on a class simply blocks a percent of the traffic from that request class and for beneficial effects to be realized on other classes there must be commensurate demands from those classes. For a highly skewed request mix of (20,20,60), blocking a large portion of class 3 requests will only marginally benefit class 1 or 2 requests. In the next experiment, we show how downgrading a blocked high bitrate request to a low bitrate service enhances the VoD performance.

3) *Performance With Downgrade Option:* In this experiment, we explore the option of downgrading a blocked request from class 2 at 8 Mb/s and class 3 at 16 Mb/s to a class 1 request at 4 Mb/s in the context of the redirect policy. Class

definition is based on request playback rate and remains the same from previous experiment. Several scenarios where we combine downgrade option with different threshold values on class 3 requests are presented. The request arrival rate is varied between 500 and 1000 requests per hour. A threshold of up to 30% and 60% capacity for the high bitrate requests are enforced along with a downgrade option. The performance metric used is the overall blocking probability for the VoD system. The results are shown for a highly skewed request mix of (20,20,60).

In Fig. 7, blocking probability of the overall VoD system is shown as a function of varying load under various options: 1) downgrade with 30% threshold; 2) downgrade with 60% threshold; 3) downgrade with no threshold. These results are compared with three other scenarios: no threshold with no downgrade option and 30% and 60% threshold with no downgrade option. Under 30% threshold, for example, only 30% of the capacity is allowed for high bitrate requests which means that a higher number of high bitrate requests are downgraded than compared to a 60% threshold or no threshold. Therefore, the downgrade option with a threshold at 30% has the least blocking because a large percentage of the blocked requests are now converted to class 1 requests for service. The other two scenarios namely the 60% threshold with downgrade and no threshold with downgrade, show similar performance but differ drastically from the 30% threshold with downgrade option. If there was no downgrade option at 30% or 60% threshold for a highly skewed request mix such as (20,20,60), it simply means that a large portion of the request population was blocked with only marginal improvement in blocking performance for class 1 and class 2. This result is discussed in the previous experiment and is illustrated by the top three curves in Fig. 7 which indicates unacceptable high overall blocking for the VoD system. This underscores the importance of the downgrade option for highly skewed request mixes under high threshold restrictions.

When comparing the three scenarios without the downgrade option (the top three curves in Fig. 7), we note that the option with 30% threshold has the worst blocking performance. The two scenarios, 60% threshold and no threshold, show comparable performance. In fact, scenario with 60% threshold performs better than no threshold scenario at high loads as shown in Fig. 7. The reason for this switch over is that with 60% threshold restrictions on class 3 requests, more capacity is available to

class 1 and class 2 requests which have appreciable demand at high loads.

4) *Policy Guidelines:* The significance of the analysis and the results presented in the previous sections comes from the fact that similar methodology can be employed by a service provider to set policy guidelines. The importance of threshold-based admission control as analyzed in this paper is that we can clearly associate the effects of different threshold levels with the resulting blocking performance. Generally, for threshold-based restriction to work well there should be a sizeable demand from all request classes. If it is a highly skewed request mix then restricting the request class that has the most demand will only result in a marginal benefit to other classes. This is where the downgrade option will help in setting appropriate policies and ensuring superior performance. With the downgrade option and threshold restriction, the service provider can actually tailor the highly skewed request mix into a workload that the VoD system can handle. Notwithstanding the obvious conclusion that the downgrade option will result in better performance, the more important observation from this paper is that it gives the service provider the flexibility to modify an incoming request mix into a workload that the VoD system can handle. Sizing and resizing the VoD system in response to fluctuating demand is not an acceptable solution and the merit of the study presented in this paper is that it provides a policy guideline to the problem of resource allocation and performance.

VI. CONCLUSION

In this paper, we analyzed the performance of a VoD system in the context of differential treatment of video requests. We presented a methodology using which we can distinguish between the merits of different classes of requests and incorporate that in a standard admission control policy. Through analytical and simulation means we showed that threshold-based admission control is a useful tool for manipulating performance for some of the request classes. We showed that enforcing threshold restrictions with the option of downgrading blocked requests in a multirate service environment results in improved performance at the same time providing different levels of QoS. This work has important implications in the areas of system design and signaling protocols. We addressed the issue of admission control and request handling with a broad focus on a distributed VoD architecture and showed how performance can be improved. We showed that redirection based request handling policies along with the option of downgraded service and threshold restrictions have superior performance. This important result points to the need for including such features in signaling protocols for distributed VoD systems.

APPENDIX PROOF OF THEOREM 1

Let \bar{e}_k be (1,0) or (0,1), where the 1 is in the k th position. Under equilibrium conditions, we have

$$\sum_{(n_1, n_2) \in \Theta(i)} P((n_1, n_2) - \bar{e}_k) = P(I = i - 1) - P(I = i - 1, n_k = l_k). \quad (16)$$

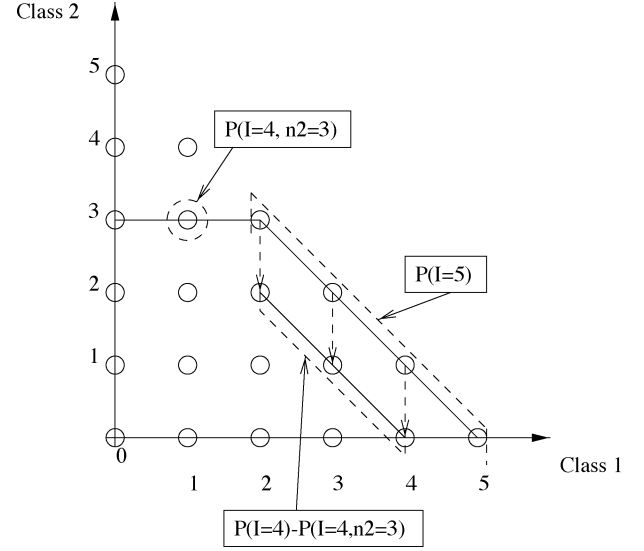


Fig. 8. One arrival backward in the 2nd dimension.

Fig. 8 illustrates (16) for class 2 requests. The left-hand side (LHS) indicates the set of states belonging to $\Theta(5)$ and the state probabilities from which we can get to the states in $\Theta(4)$ and their probabilities by moving one arrival backward for class 2. But this set of states in $\Theta(4)$ does not include the blocking state as indicated on the right-hand side (RHS)

$$\sum_{(n_1, n_2) \in \Theta(5)} P((n_1, n_2) - \bar{e}_2) = P(I = 4) - P(I = 4, n_2 = 3)$$

Consider resources used $I = \sum_{k=1}^2 n_k$ and multiplying (6) with i and substituting for i and rearranging on the RHS, we have

$$iq_i = \sum_{k=1}^2 \sum_{(n_1, n_2) \in \Theta(i)} n_k P(n_1, n_2).$$

Substituting from the balance equation, $n_k P(n_1, n_2) = \rho_k P((n_1, n_2) - \bar{e}_k)$ we have

$$iq_i = \sum_{k=1}^2 \sum_{(n_1, n_2) \in \Theta(i)} \rho_k P((n_1, n_2) - \bar{e}_k)$$

$$iq_i = \sum_{k=1}^2 \rho_k \sum_{(n_1, n_2) \in \Theta(i)} P((n_1, n_2) - \bar{e}_k)$$

Substituting from (16) for $\sum_{(n_1, n_2) \in \Theta(i)} P((n_1, n_2) - \bar{e}_k)$ we have

$$iq_i = \sum_{k=1}^2 \rho_k [P(I = i - 1) - P(I = i - 1, n_k = l_k)].$$

Multiplying with G and using (7) and moving i to RHS we have the recursion in Theorem 1

$$G(i) = \frac{1}{i} \sum_{k=1}^2 \rho_k [G(i - 1) - B_k(i - 1)] \quad i = 1, \dots, C$$

and $G(i) = 1$ for $i = 0$ and 0 for negative i . \square

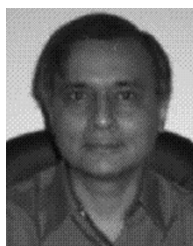
REFERENCES

- [1] L. Kleinrock, *Queueing Systems: Volume I, Theory*. New York: Wiley, 1975.
- [2] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control—the single node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.
- [3] R. Zimmerman and K. Fu, "Comprehensive statistical admission control for streaming media servers," in *Proc. ACM Multimedia Conf.*, 2003, pp. 75–85.
- [4] A. L. N. Reddy and J. Wyllie, "Disk scheduling in a multimedia I/O system," in *Proc. ACM Multimedia Conf.*, 1993, pp. 225–233.
- [5] E. Chang and A. Zakhor, "Admission control and data placement for VBR video servers," in *Proc. 1st IEEE Int. Conf. Image Processing*, 1994, pp. 278–282.
- [6] H. M. Vin, P. Goyal, A. Goyal, and A. Goyal, "A statistical admission control algorithm for multimedia servers," in *Proc. ACM Multimedia Conf.*, 1994, pp. 33–40.
- [7] S. Bakiras and V. O. K. Li, "Maximizing the number of users in an interactive video-on-demand system," *IEEE Trans. Broadcast.*, vol. 48, no. 4, pp. 281–292, Dec. 2002.
- [8] I. R. Chen and C. M. Chen, "Threshold-based admission control policies for multimedia servers," *Computer J.*, vol. 39, no. 9, pp. 757–766, 1996.
- [9] J. M. Aein, "A multi-user-class, blocked-calls-cleared demand access model," *IEEE Trans. Commun.*, vol. COM-26, no. 3, pp. 378–385, Mar. 1978.
- [10] K. W. Ross and D. H. K. Tsang, "The stochastic knapsack problem," *IEEE Trans. Commun.*, vol. 37, no. 7, pp. 740–747, Jul. 1989.
- [11] S.-H. Chan and F. A. Tobagi, "Threshold-based admission policies for video services," in *Proc. Global Telecommunications Conf.*, 1999, pp. 2076–2080.
- [12] P. Mundur, R. Simon, and A. Sood, "End-to-end analysis of distributed video-on-demand systems," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 129–141, Feb. 2004.
- [13] 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.
- [14] J. S. Kaufman, "Blocking in a shared resource environment," *IEEE Trans. Commun.*, vol. COM-29, no. 10, pp. 1474–1481, Oct. 1981.
- [15] J. W. Roberts, "A service system with heterogenous user requirements—application to multiservices telecommunications systems," in *Proc. Performance Data Communications Systems Conf.*, 1981, pp. 423–431.
- [16] J. R. Buzen, "Computational algorithms for closed queueing networks with exponential servers," *Commun. ACM*, vol. 16, no. 9, pp. 527–531, 1973.
- [17] D. H. K. Tsang and K. W. Ross, "Algorithms to determine exact blocking probabilities for multirate tree networks," *IEEE Trans. Commun.*, vol. 38, no. 8, pp. 1266–1271, Aug. 1990.
- [18] H. Schwetman, *CSIMIS User's Manual*: Mesquite Software, Inc., 1998.
- [19] G. K. Zipf, *Human Behavior and the Principle of Least Effort*. Reading, MA: Addison Wesley, 1949.
- [20] P. V. Mundur, "An integrated approach to end-to-end analysis of distributed video-on-demand systems," Ph.D. dissertation, George Mason University, Fairfax, VA, 2000.



Padmavathi Mundur received the M.E. degree in systems engineering from the University of Virginia, Charlottesville, VA, in 1990 and the Ph.D. degree in information technology from George Mason University, Fairfax, VA, in 2000.

She is currently an Assistant Professor in the Department of Computer Science and Electrical Engineering at University of Maryland, Baltimore County, Baltimore. Her research interests include distributed systems, multimedia networking, and analytical performance modeling and resource allocation techniques. She has served on the program committees of ICDCS 2004, ICME 2004, and been a reviewer for National Science Foundation panels, journals, and conferences.



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

He is the Chair and Professor of Computer Science, Department of Computer Science, 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. 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.



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 and the University of Pittsburgh Medical Center, and spent a summer at Hewlett-Packard Laboratories, Palo Alto, CA. His research interests are in networks, real-time and distributed systems, and simulation methodologies. He has published over 60 peer-reviewed conference and journal papers in these areas. He 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.