Integrated Admission Control in Hierarchical Video-on-Demand Systems

Padmavathi Mundur, Robert Simon and Arun Sood
Department of Computer Science
George Mason University
Fairfax, VA 22030
{pmundur,simon,asood}@cs.gmu.edu

Abstract

In this paper, we develop a unified model of a hierarchical Video-on-Demand(VoD) system by integrating the storage and the network subsystems. Rather than restricting the analysis to an isolated subsystem the performance of the VoD system is analyzed as an end-to-end system. On a system-wide basis, request handling and admission control policies are designed to minimize global performance metrics. Through our simulation, we compare different request handling policies and show that a hierarchical VoD architecture with request handling that allows retrials at more than one resource will minimize overall blocking.

Keywords: *Network and Resource Management, Distributed VoD, QoS Control*

1. Introduction

The topic of this paper is the development of a unified model for a hierarchical Video-on-Demand(VoD) system. The model integrates video server retrieval and network transfer mechanisms by relating buffer management at the server with rate-based scheduling at the network. The significance of this integrated analysis is the development of a method for designing and measuring the effectiveness of end-to-end admission control and request handling policies. Contributions of this paper are threefold. First, a unified model for a hierarchical VoD architecture consisting of clusters of video servers and network elements is developed. Second, global performance metrics are developed and used in evaluating the performance of the VoD system. Third, request handling policies and admission control techniques that result in an efficient use of multiple resources are designed and evaluated.

Much of previous research is focused on topics related to a single video server design such as disk striping, video block placement, and admission control at the level of disks and disk groups [3], [9], [10]. Research in real-time networks, analyzing network conditions for guaranteed services has been discussed in [4], [7]. Research in distributed VoD systems focuses on load balancing schemes based on replication and placement techniques [1], [2], [10]. In [1], the focus is on movie placement and load balancing in a distributed VoD architecture. Their placement scheme assigns movies to each level of an r-ary treelike network topology so that expected demand adds up to the fraction of the traffic supported by the link at that level. The bandwidth capacity is used to characterize the network in the load balancing scheme in [2]. Our paper is different from [1] and [2] in that load balancing is not the primary focus. In this paper, we explicitly model using a real-time network for general purpose request handling in a hierarchical VoD system.

2. Hierarchical VoD system architecture

Our hierarchical VoD system architecture consists of local and remote sites. Each site is characterized by a cluster of video servers. The local cluster serves users over a local distribution network such as an ATM LAN. We assume that there are sufficient network resources at the local cluster to deliver videos to the users. The remote site may be archival in nature, providing a permanent repository for all videos. Remote servers also provide video delivery service over high speed networks.

A typical organization of local and remote clusters is shown in Figure 1. The hierarchy of clusters of servers and networks results in a scalable VoD system by providing multiple resources to the user population; if a request cannot be serviced from the local site, it may be directed to other remote sites. The set-top boxes at the client site provide the decoding and display functionality for delivered videos from the video servers.

End-to-end admission control

Real-time requirements for multimedia data transfer dictate a tight coordination among server and network components for remote transfers. Resources must be available at

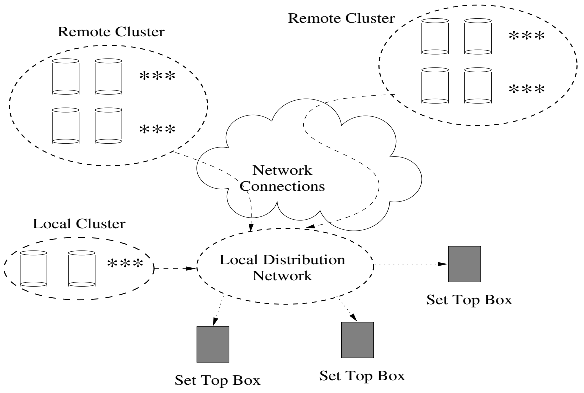

Figure 1. Hierarchical VoD architecture

each element before a new request is admitted into the system. At the server, an admission control algorithm checks if the required disk bandwidth is available. At the network, a routing *path* must be set up and resources reserved along that path. The required bandwidth must be guaranteed for a bounded delay requirement. The unified admission control algorithm must reject the arriving request if any of these requirements cannot be met.

Server and network model

The server model consists of a storage architecture based on high capacity and high bandwidth storage such as RAID3 or RAID5. The data is transferred to the client periodically; that is, the requests are processed in *rounds* [3]. Within each round, a disk scheduling algorithm such as Scan EDF or Grouped Sweeping Scheme [3] must be employed to provide guaranteed real-time retrieval. One of the key aspects of our unified model is the following. The buffer size at the client's set-top box is the amount of data retrieved at the server and sent over the network to the client in each round. The round length itself is determined by the duration of the data in the client's buffer played back at a known rate called the playback rate. We prevent starvation at the client buffer by delivering the new data before the end of the round length.

Servicing requests from remote clusters involves data transfer over the network. Figure 2 illustrates the stages involved in the remote transfer. In our network model, the double-buffer at the server acts as the packet source for the *traffic regulator*. The regulator is used to monitor the flow of traffic into the network. We assume that a (b r p M) regulator [4] exists at the server. Each request is modeled as a flow passing through this regulator into the network. We assume *Weighted Fair Queueing* (WFQ) scheduling in the network. In [7], 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. We use the WFQ delay bound and invert

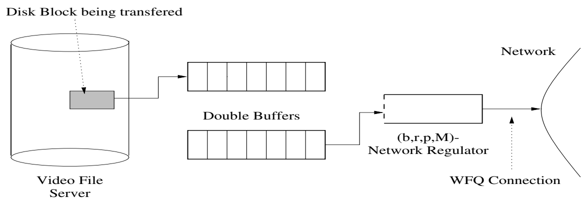

Figure 2. Double buffer at the server with fair queueing network

Table 1. Model parameters

B	Buffer size at the receiver (client set-top box)
R_{pl}	Playback rate per request (at the client)
R_{nw}	Maximum bandwidth per network link
R_d	Overall disk bandwidth
R_{res}	Reserved rate at the network per request
T_{round}	Round length
n	Maximum number of requests at the server
$n_{current}$	Number of requests currently being served

it to find reserved rates at the network in the next section.

3. Analytical model for remote service

In this section, we develop a model for remote transfers by relating the round length parameter at the server with the reserved rate at the network with the buffer size at the client. The model parameters are shown in Table 1.

As illustrated in Figure 2, the size of the buffer in the double buffer scheme corresponds to the total data retrieved in a round length. The round length is the duration of the data retrieved per request at the client site. The round length, T_{round}, for the server is determined by:

$T_{round} = \frac{B}{R_{pl}}$

The maximum number of requests, n, serviceable at the server is derived as follows. The time to transfer the amount of data retrieved per round is bounded by the round length, that is, $\frac{nB}{R_d} \leq T_{round} = \frac{B}{R_{pl}}$. The maximum number of requests that can be serviced at the server, n, is: $n \leq \left\lfloor \frac{R_d}{R_{pl}} \right\rfloor$.

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

$n_{current} + 1 \leq n$

For videos which must use a network connection, the minimum reserved rate, R_{res}, at the network required to prevent starvation at the client site is given by:

$R_{res} \geq \frac{B}{T_{round}} = R_{pl}$

The reserved bandwidth cannot be greater than the overall bandwidth of each link on the path, $R_{res} \leq R_{nw}$ assuming homogenous links. Therefore, the bounds on reserved rate at the network, R_{res}, are: $R_{pl} \leq R_{res} \leq R_{nw}$.

By using a (b r p M) regulator, the end-to-end delay bound $\bar{D}$ is given by [4]:

$$\bar{D} = \frac{b-M}{R_{res}}\frac{p-R_{res}}{p-r} + \frac{M+C_{tot}}{R_{res}} + D_{tot} \quad \text{if } R_{res} < p$$
$$\bar{D} = \frac{M+C_{tot}}{p} + D_{tot} \quad \text{otherwise}$$

Since R_{res} is almost always less than p, the maximum rate of the network interface card, we only consider the first case. At the (b r p M) regulator, the depth of the token bucket, b, is the same as the data retrieved per request per round, B. The rate at the token bucket is equal to the reserved rate at the network, $r = R_{res}$. After substituting for b and r and some simplification, the end-to-end delay is:

$$\bar{D} = \frac{B}{R_{res}} + \frac{C_{tot}}{R_{res}} + D_{tot}$$

where $C_{tot} = \sum_{j=1}^{J} C_j$ and $D_{tot} = \sum_{j=1}^{J} D_j$. The values for C_j and D_j depend on the scheduling mechanism implemented at the network. Using WFQ, the maximum delay bound per packet for a flow with a guaranteed rate of R_{res} and routed on a path that has J links is given by [7]:

$$\bar{D} = \frac{B}{R_{res}} + \frac{(J-1)M}{R_{res}} + \sum_{j=1}^{J} \frac{M_{max}}{A_j} \tag{1}$$

where M is the maximum size of the packet for the flow, M_{max} is the maximum size of a packet permitted in the network (MTU), and A_j is the overall bandwidth on link j.

Using (1), we now derive R_{res} corresponding to an input delay bound $\bar{D}$. This required minimum rate, $R_{res} \geq R_{pl}$, is used in the admission control test before admitting a new request on the network.

$$R_{res} = \frac{B + (J-1)M}{\bar{D} - \sum_{j=1}^{J} M_{max}/A_j} \tag{2}$$

Admission control tests for new flow requests is run on a link by link basis over the routing path. Suppose that link j has currently accepted T flows $1, 2, \ldots, T$. The i^{th} flow has a reserved rate of R_{res}^{i} computed using (2). If the *new* flow requests a rate of R_{res}^{new}, it will be admitted only if the following relation holds:

$\sum_{i=1}^{T} R_{res}^{i} + R_{res}^{new} \leq A_j$

4. End-to-end performance analysis

In this section, we describe global performance metrics and request handling and admission control policies. The arriving request is routed to either the local cluster or the remote clusters based on policy restrictions and availability constraints. Since storage is limited on the local cluster, only popular videos are stored on it. As a result, requests for the less popular videos will be *directly* served from the remote clusters. A request that gets rejected in the admission control test is said to be *blocked*. The number of blocked requests forms the basis for developing the global performance metric.

Global performance metric

The VoD system offers a video collection of V videos. The average request arrival rate for k^{th} video is λ_k. The blocking probability, b_k for k^{th} video is defined as the ratio of the number of blocked requests to arrivals.

Overall blocking rate: The overall blocking rate indicates the number of arriving requests per time unit at the VoD system that will get blocked. The overall blocking rate is defined by:

$BR_{vod} = \sum_{k=1}^{V} \lambda_k b_k$

Class-based overall blocking: The video collection is divided into classes, each class containing a subset of the video collection. Class-based blocking for various policies can be measured.

Resource blocking: Since there are multiple resources in our VoD system, measuring blocking at each resource indicates the level of activity at that resource.

Request handling policies

Request handling policies are defined to determine where to establish connection for the new request in our hierarchical architecture. The policies are designed based on three criteria: whether remote service is available; whether redirection to other clusters for blocked requests is permitted; whether the incoming traffic is first split between local and remote clusters. Implementing a split policy with no redirection for blocked requests is the simplest, but efficient splits may be hard to determine. A policy based on complete or some redirection has higher implementation overhead, because of the multiple checks per blocked request.

Threshold-based admission control policies

During admission control, in addition to capacity restriction, threshold-based restrictions may be imposed for some videos. Threshold-based admission control policies are known to result in better performance than *complete sharing* policies [6]. Threshold policies restrict admission at the server so that a request for a video over its threshold limit will not be admitted to the system even if the capacity is available. Threshold-based admission control may be enforced in combination with request handling policies. Because of the large number of videos in the collection, the videos are aggregated into classes. Threshold restrictions are imposed on classes which are inherited by the videos in those classes. A threshold of 0.6 means that only 60% of the request handling capacity of a resource is available to that class and hence, to all the videos in that class.

5. Simulation

In this section, we provide details on simulation data as well as the results of the performance evaluation. We conducted the simulation using the CSIM simulation package [8]. We use the metrics from Section 4 to evaluate the performance of the VoD system.

Simulation setup

The specific hierarchical architecture simulated consists of three levels: local cluster, remote cluster1 via network1, and remote cluster2 via network2. Each local cluster server is modeled after a 16 Gigabytes(GB) RAID-3 that can simultaneously serve 22 request streams of 4 megabits per second (Mbps)[9]. The number of servers at the local cluster is 10. Adequate storage and stream handling capacities are assumed at the servers in the remote clusters and requests are not blocked. This is not a limitation of our model; it is done to simplify the presentation of results. Link capacities are 622 Mb/s and the maximum packet size and MTU are 1500 bytes [4]. The number of hops to remote cluster1 is 3 and to remote cluster2 is 6. The burst size of 100 kilobytes (kB) for the token bucket regulator at the servers is also the buffer size at the clients' set-top boxes. Input end-to-end delay is assumed to be 300 ms. The reserved rate at the network corresponding to the input delay is calculated using (2). The reserved rate is at least as large as the playback rate.

The access patterns to each video depend on the popularity of the video. We use a Zipf distribution to find the probability of choosing k^{th} video from a collection of V videos [10]. The probability of choosing the k^{th} most popular video from a collection of V videos, is found by Z/k where $Z = 1/(1 + 1/2 + 1/3 + \ldots + 1/V)$. The order in which they appear is also an indication of their popularity, the first video being the most popular and the 15^{th} video the least popular. Using the Zipf distribution, if the overall arrival rate of requests to the VoD system is λ, the arrival rate of 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 is used as the basis for replication and placement of videos [5].

In our simulation, the arrival process for each request is assumed to be a Poisson process with exponential interarrival times. The range of values for arrival rate, λ, used in the simulation is between 200 and 380 requests per hour. The total number of hours simulated is 10000. The number of videos in the video collection is 15. The service times in hours range from 0.68 to 2.03. The size of videos computed using a playback rate of 4 Mbps ranges from 1.23 GB to 3.66 GB.

Since we have an ordered list of videos based on popularity, we have defined class1 as the top 20% of the videos; class3 as the bottom 20% and the rest in the middle as class2. We limit our analysis to three classes, but the number of classes and the video composition in each class can be varied for an extended analysis. In our simulation, the class2 and class3 threshold values used are in the range of 0% to 20%.

The specific policies evaluated are:

Direct - No remote service available.

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-Redirect56 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 service 60% and the 40% goes to remote cluster2. 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 the blocked requests at local cluster are not redirected to remote clusters.

Performance evaluation

We show in the following analysis these major results:

(1) Hierarchical organization provides storage for less popular and archival videos at remote sites in addition to providing supplemental request handling capacity to minimize overall blocking.

(2) The performance of redirect policy is strikingly better than the other policies despite having the highest path establishment overhead. The difficulty in finding a split that minimizes overall blocking rate makes the split based policies less attractive, in spite of their simple implementation.

(3) Threshold-based admission reduces blocking rates for the more popular videos, though at the cost of increasing blocking for video classes that have threshold restrictions.

(4) An adaptive request handling policy where one policy replaces another based on load conditions will enhance the overall performance of the VoD system.

Extended results using the analytical framework from Section 3 are presented in [5] where we show that as VoD system size is scaled up in response to increasing demand, the overall blocking percentage remains the same. We also show that the performance of these policies is insensitive to holding times.

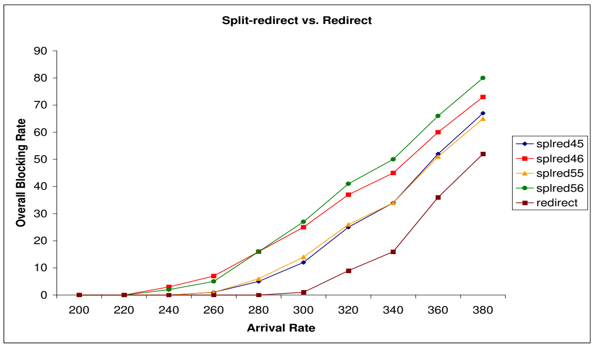

Figure 3. Blocking rate: Redirect and Split-Redirect

In Figures 3 and 4 performance of redirect policy is compared with various combinations of split-redirect and split policies. Redirect policy has the minimum overall blocking rate in both cases. The redirect policy gives maximum utilization at each resource because remote service is delivered only when the local cluster cannot handle the load. Among the split policies, the 45 split shows the minimum overall blocking rate for both split-redirect and split policies. The rest of the analysis with respect to split policies is based on the 45 split.

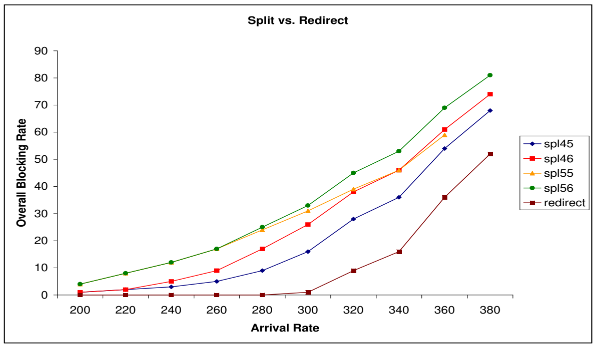

Figure 4. Blocking rate: Redirect and Split

Figure 5 shows the difference in performance of the VoD system with and without remote service. Without remote service the system exhibits an overall blocking of 38% at the lowest load of 200 arrivals per hour. However, the performance improves drastically with remote service from the

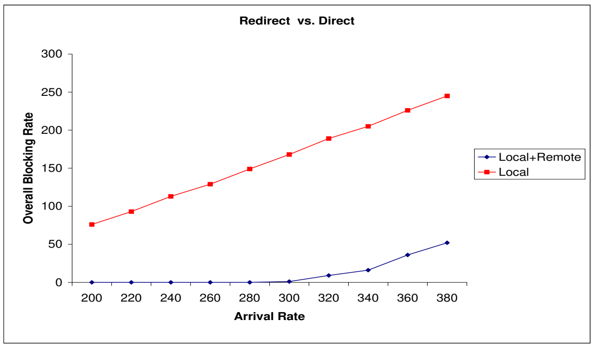

Figure 5. Blocking rate with and without remote service

two remote clusters. Even at a high load of 380 arrivals per hour, the overall blocking is at about 13%. So the hierarchical architecture in addition to providing more storage, also provides supplemental request handling capacity.

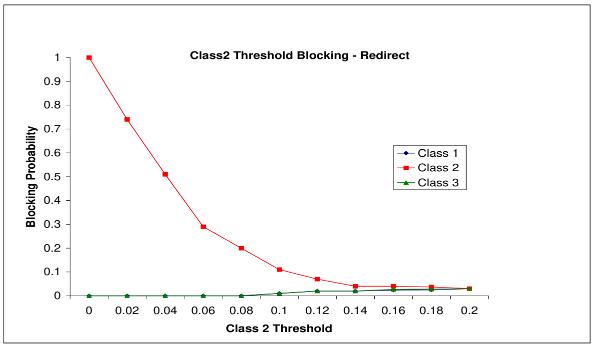

Figure 6. Class 2 threshold effect on blocking - Redirect

In Figures 6 and 7 the effect of threshold on blocking for redirect policy is shown. The threshold experiments are done at a moderate load of $\lambda = 320$ requests per hour. Class2 and class3 consisting of the less popular videos are subjected to various thresholds. In Figure 6, class2 threshold results in 0% to 3% blocking for class1 and class3 videos. Blocking for class2 is 100% at 0 threshold, meaning that no requests of class2 are allowed into the system. Blocking levels off at 3% along with the other two classes around 20% threshold when the threshold has no effect since the arrival rate is below the threshold anyway. In Figure 7 the effect of class3 threshold on class1 and class2 blocking is shown. The same kind of behavior is observed with the split-redirect and split policies.

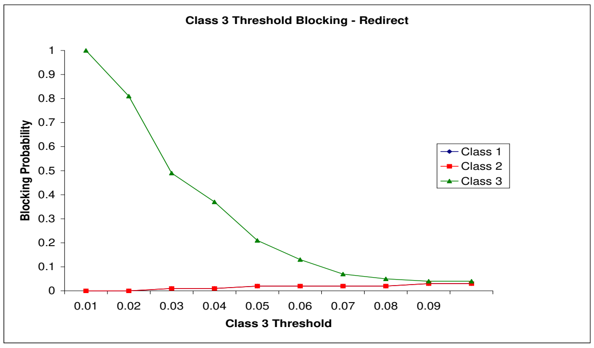

Figure 7. Class 3 threshold effect on blocking - Redirect

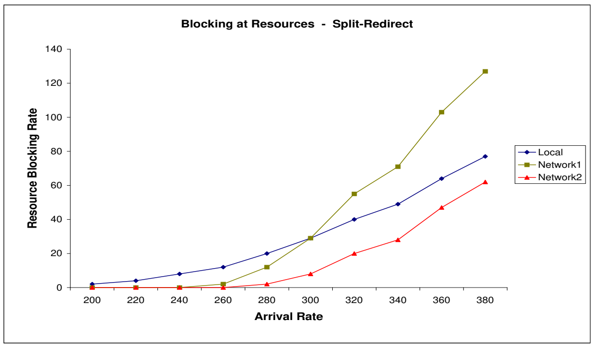

Figure 8. Blocking rate at resources - Split-Redirect45

Figure 8 indicates activity local to the resources in the VoD system for split-redirect45 policy. Due to redirected traffic from local cluster, blocking at network1 increases at an unacceptable rate for a load of 280 or more requests per hour. At that point, with an adaptive policy we can decide to stop redirecting the blocked traffic from the local cluster. If there is no redirection the situation reduces to the split45 policy and blocking at network1 is similar to network2. In the absence of the redirect policy, we can toggle between split-redirect and split policies depending on the workload.

6. Conclusion

In this paper, we presented a unified model of a distributed VoD system. We developed global performance metrics to evaluate request handling and admission control policies. Different request handling policies are compared and the redirect policy is shown to have the minimum blocking rate. We have shown that rather than load sharing, retrials at more than one resource result in superior overall system performance and may be worth the additional complexity in establishing the path.

References

[1] C. Bisdikian and B. Patel. Issues on movie allocation in distributed Video-on-Demand systems. In *Proceedings of the IEEE ICC*, pages 250–255. IEEE Press, 1995.

[2] D. W. Brubeck and L. A. Rowe. Hierarchical storage management in a distributed VoD system. *IEEE Multimedia*, pages 37–47, Fall 1996.

[3] D. J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L.A. Rowe. Multimedia storage servers: A tutorial. *IEEE Computer*, 28(5):40–49, 1995.

[4] L. Georgiadis, R. Guerin, V. Peris, and R. Rajan. Efficient support of delay and rate guarantees in an internet. In *ACM SIGCOMM*, pages 106–116, August 1996.

[5] P. Mundur, R. Simon, and A. Sood. End-to-end request handling in distributed Video-on-Demand systems. In *Proceedings of the CNDS Conference (Society for Computer Simulation)*, pages 151–157, 1999.

[6] P. Mundur, A. Sood, and R. Simon. Threshold-based admission control for multi-class Video-on-Demand systems. In *Proceedings of the IEEE IPCCC*, pages 154–160, February 1998.

[7] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control - the multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, 1994.

[8] H. Schwetman. *CSIM18 User's Manual*. Mesquite Software, Inc.

[9] Y. Wang, J. Liu, D. Du, and J. Hsieh. Efficient video file allocation schemes for video-on-demand services. *Multimedia Systems*, 5(5):283–296, September 1997.

[10] J. Wolf, P. Yu, and H. Shachnai. Disk load balancing for Video-on-Demand systems. *Multimedia Systems*, 5(5):358–370, September 1997.