Optimized Caching in Systems with Heterogeneous Client Populations[②]

Derek L. Eager — University of Saskatchewan, Saskatoon, SK Canada S7N 5A9

Michael C. Ferris, Mary K. Vernon — University of Wisconsin - Madison, Madison, WI 53706-1685

***Abstract*-- Supporting on-demand access to large widely shared data, such as popular video objects, requires effective use of regional (proxy) servers that store some of the data close to the clients. The proxy caching problem is more complex in the context of continuous media files because of the need to consider bandwidth as well as storage constraints at the proxy servers, and because of the bandwidth sharing possibilities provided by recently proposed multicast delivery techniques. This paper develops new models fo determining optimal proxy cache content in such environments. Specifically, the models developed here provide insights for heterogeneous systems in which the proxy servers have differing client populations and server capabilities. The new results show that (1) in comparison to previous results for systems with homogeneous proxy workloads, it is even more often cost-effective to cache just the initial segments of many files, rather than the complete data for fewer of the most popular files, (2) to minimize total delivery cost, even in systems with quite strong heterogeneous features, it is often best for all proxy servers to store *similar* data, rather than to closely tailor each proxy cache content according to local client preferrences and server characteristics, (3) when minimizing total delivery cost, a (group of) regional server(s) with a distinct client workload can sometimes influence the data stored by the rest of the servers in unexpected ways, and (4) when minimizing the delivery cost for clients of an individual proxy server, the data to be stored at the proxy server may be quite different than the "socially optimal" set that would minimize total delivery cost to all clients.**

***Index terms*-- multimedia delivery techniques, Web servers, proxy caching, optimal cache content.**

A. INTRODUCTION

In systems that support on-demand access to widely-shared data, delivery cost can be greatly reduced through the use of regional (or proxy) servers that store some of the data close to the requesting clients. For popular objects, such as popular news clips, product advertisements, distance education content, and the like, significant savings in both server and network bandwidth can also be achieved through the use of multicast (or broadcast) delivery to multiple clients simultaneously. For large objects, such as video data, it is also advantageous to divide the file into segments, so that clients can receive and buffer some segments ahead of need, at a bandwidth (i.e., cost) savings because the multicast of those segments is already scheduled for clients who requested the object earlier [22, 23, 1, 16, 11, 15, 12].

This paper addresses the question of which large popular widely shared data should be stored at the regional/proxy servers, in the context of systems containing one or more shared remote servers, multicast delivery, segmented data objects, and multiple heterogeneous proxy servers. Our goal is to achieve *broad insights into proxy caching strategies* for such systems, rather than to compute the precise cache contents for any particular system. We assume the system uses the recently proposed partitioned dynamic skyscraper delivery technique [12] outlined in Section B, although our results are also applicable to other multicast delivery techniques.

Most prior work on Web caching as well as distributed video-on-demand (VOD) architectures has largely focussed on determining on which server(s) each entire object should be allocated, so as to optimize system cost/performance [19, 2, 4, 7, 6]. Other related work has concerned strategies for dynamically caching intervals of data from continuous media World-Wide Web objects, so as to satisfy multiple requests that arrive close in time [21]. However, these previous papers on whole object placement and interval caching have not considered the impact of *shared delivery of popular objects* that is enabled by multicast delivery techniques. In particular, there is a new trade-off between caching the objects that are requested most frequently and caching less popular objects that have lower cost sharing when delivered from the remote server. Furthermore, for segmented delivery techniques, the initial segments for a given object are smaller and have more frequent multicasts (i.e., higher bandwidth requirement per byte) with fewer clients per multicast (i.e., less cost sharing of remote delivery) than the larger later segments of the object. This leads to another key cost trade-off between caching the entire data for particular objects or caching the initial segments of many more objects.[1]

Models that include client cost-sharing, as well as server costs for the new multicast delivery techniques, are needed to evaluate the above proxy design trade-offs. A previous optimization model [12] determines what data should be stored at *homogeneous* proxy servers to optimize cost/performance, under specified bandwidth and storage constraints at the proxy, assuming partitioned dynamic skyscraper multicast delivery and the capability to cache all, none, or a specified number of initial segments of each object.

[②] This work was partially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Grant OGP-0000264, by the Air Force Office of Scientific Research (AFOSR) under Grant F49620-98-1-0417, and by the National Science Foundation (NSF) under Grant CCR-9975044.

[1] Recent work has established that caching just the initial segments of objects has a number of other advantages [12, 20], including hiding the latency of communication with a remote server, and facilitating workahead smoothing of variable-bit-rate video.

We extend this previous model to study systems in which the proxy servers differ with respect to their bandwidth and storage capacities, and with respect to their client workloads.

Heterogeneity is an important factor because it occurs in many practical systems and it introduces a *new tension* between (a) tailoring the data stored at each proxy according to the local client workload, and (b) maximizing uniformity in proxy cache contents so as to achieve the greatest possible sharing of multicasts of uncached items from the remote server. In other words, heterogeneity may cause a divergence between the globally optimal cache configurations, and what is individually optimal for each regional site. A key goal of this work is to create new optimization models that allow us to gain insight into how this tension is resolved for various kinds of heterogeneity.

A key challenge is to create optimization models that are tractable and yet allow us to gain the desired insights for heterogeneous systems. We have created two such models, which are described in Section C. The models are applied in Section D to study proxy server data caching strategies, and the impact of heterogeneity. The insights and design principles derived from the results include:

1. It is often more cost-effective to store just initial segments rather than entire objects at the proxy servers. Heterogeneity increases this tendency.
2. Even in systems with quite strong heterogeneous features, caching similar data at all of the proxy servers often results in a lower total delivery cost, than does tailoring each proxy content according to the local client workload and server capabilities.
3. When minimizing total delivery cost, a (group of) proxy server(s) with a distinct client workload can sometimes influence the data cached by the other proxy servers in unexpected ways.
4. When minimizing the delivery cost for the clients of a given proxy server, the optimal set of data to store at that server may be quite different than the "socially optimal" set that minimizes total delivery cost to all clients.

Section E summarizes the contributions of this work and discusses on-going and future research.

B. BACKGROUND

Section B.1 describes segmented multicast delivery techniques and the partitioned dynamic skyscraper delivery technique that is assumed in the optimization models developed in this paper. Section B.2 reviews the previous optimization model for determining optimal cache content in partitioned dynamic skyscraper systems with homogeneous proxy servers.

Throughout the remainder of the paper, the term “channel” is used to denote the collection of server and/or network resources required to support a single (multicast) transmission stream, at the required data delivery rate.

B.1 Segmented Multicast Delivery

The conventional approach to on-demand delivery of video and other continuous media objects is to allocate a new delivery stream (or channel) for each client request.

A simple mechanism for conserving the bandwidth required for a given very popular object entails reserving some number of streams or channels for periodic multicasts of the object. The starting times on the reserved channels are staggered so as to bound the maximum time that a client needs to wait until a new transmission of the requested object commences. Clients that make closely-spaced requests to the same object batch together while waiting, and are served by the same multicast transmission.

Recently proposed segmented multicast delivery techniques [22, 23, 1, 16, 15, 11, 12] achieve a significantly better bandwidth savings by dividing each hot object into fixed increasing-sized segments, and employing transmission schedules in which the smaller initial segments are multicast more frequently than the remaining larger segments. By frequently multicasting the first segment, the time that a client must wait to receive the segment (and thus, for video, to commence playback), is reduced. Multicasting the larger segments less frequently reduces bandwidth usage. The structure of the transmission schedules ensures that segments can always be received by the time they're needed for playback. Since the larger segments are multicast less frequently, clients must be prepared to receive such segments ahead of when they are needed, batching together with other clients that may be at quite different playout points. This implies increased client requirements for buffer space and for reception bandwidth, as multiple segments may need to be received concurrently.

Most of the segmentation-based delivery techniques employ static transmission schedules and are thus only applicable to objects that are steadily extremely popular. An exception is the dynamic skyscraper technique [11], which dynamically assigns server bandwidth, in the form of multicast “transmission clusters”, for segmented delivery of objects in response to client requests. Dynamic scheduling can provide immediate service to client requests, as well as segmented multicast delivery for less popular objects, and in contexts where object popularity varies (for example, with the time of day).

The dynamic skyscraper technique was recently extended [12] to support delivery in environments that have regional (or proxy) caching. This *partitioned skyscraper* technique divides the segments of each object into two sets, one composed of the first *k* segments (the “leading segment set”) and one composed of the remaining segments (the “trailing segment set”). Separate transmission clustered are dynamically scheduled for delivery of the leading segment sets and the trailing segment sets. Scheduled delivery of the two sets is coordinated so that, for each client, jitter between playback of each set is avoided. For each object, a regional server may cache just the leading segment set, both sets of segments, or neither set.

Allocation of separate transmission clusters to the two sets of segments provides more efficient bandwidth usage (even in the absence of proxy servers) by reducing the bandwidth allocated in each leading segment cluster and by increasing the window for joining scheduled or on-going trailing segment transmissions. The loose coupling also relaxes the synchronization needed between the remote and regional server, in the case where only the leading segment set is cached at the regional server. For more information about the partitioned dynamic skyscrapter system, the reader is referred to [12].

Recent work [14] shows that the average server bandwidth required for partitioned dynamic skyscraper delivery, per object, grows asymptotically only logarithmically with the client request rate, whereas the average bandwidth required for the recently proposed Stream Tapping [5] or Grace Patching [17] multicast delivery technique grows asymptotically as the square root of the client request rate. However, the stream tapping/patching schemes also transmit initial portions of a popular continuous media object more frequently than later portions of the object. Thus, systems that use these delivery techniques would have similar trade-offs with respect to optimal proxy cache content (discussed in section 1), and similar optimal caching strategies (at the level of detail studied in section 4) as for systems that use segmented multicast techniques.

B.2. Optimization Model for Homogeneous Systems

A previous model [12] determines the proxy cache content that minimizes delivery cost for systems with (a) identical proxy server bandwidth and storage capabilities, (b) statistically homogeneous regional client workloads, (c) partitioned dynamic skyscraper delivery, and (d) equal-sized objects. For each object, the proxy servers can store either none, all, or just the leading segment set of the object. The model considers client cost sharing for the multicast delivery, as well as the relative cost of remote and proxy server resources, in determining the optimal data to store at the proxy servers. In this section we provide an overview of the model and the results that were previously obtained from the model, which provides the starting point for developing more complex models of systems with heterogeneous proxy server capabilities and workloads.

To determine delivery cost, the homogeneous model uses simple analytic estimates of the number of remote server channels (C_{remote}) and proxy server channels ($C_{regional}$) needed to support a given client workload as a function of the object segment sets that are stored at the proxy. Results given in [12] show that these server bandwidth estimates are very close to the knee of the curve of mean client waiting time versus the inverse of the number of server channels, and that the knee of the curve is typically quite sharp for systems that use multicast delivery. Furthermore, since the system can provide immediate service to client requests, the average client waiting time is typically very small near the (sharp) knee of the curve.

Table 1: **Parameters of the Homogeneous System Model**

Input	**Parameter Definition**
n	number of objects
$N_{channels}$	maximum number of channels at each proxy server
$N_{segments}$	storage capacity at each proxy server
P	number of proxy servers
β	the cost ratio of a proxy server channel and a remote server channel
λ_i	total arrival rate of requests for object i (from all regions)
k	number of segments in the leading segment set
K	total number of segments in the leading and trailing segment sets
s_j	size of the j'th segment (relative to the size of the first)
W	the largest segment size (in the trailing segment set)
Output	**Parameter Definition**
C_{remote}	number of channels needed at the remote server
$C_{regional}$	number of channels needed at the proxy server
$D_{regional}$	storage needed at each proxy server (measured in units of s_1)
θ_i^R	equals 1 if object i is stored only at the remote server; 0 otherwise
θ_i^p	equals 1 if only the leading segment set of object i is cached regionally; 0 otherwise
θ_i^r	equals 1 if object i is entirely cached regionally; 0 otherwise

Table 1 gives the model input and output parameters. Note that the last four input parameters specify the particular configuration of the partitioned dynamic skyscraper delivery system. The key system constraints in the optimization are the maximum proxy server bandwidth ($N_{channels}$), and storage capacity ($N_{segments}$). The key model outputs are the θ_i values that specify for each object i, whether the object should be fully or partially cached, or not cached, at the proxy servers.

Given that β is the ratio of costs for proxy server channels and remote server channels, and given that P is the number of proxy servers in the system, the specific homogeneous system model for optimal proxy cache content is defined as follows:

$$\min_{\theta^R, \theta^p, \theta^r} \quad C_{remote}(\theta) + P\beta C_{regional}(\theta)$$

subject to

$$C_{regional}(\theta) \le N_{channels}$$

$$D_{regional}(\theta) \le N_{segments}$$

$$\theta_i^R + \theta_i^p + \theta_i^r = 1, \quad i = 1,2,...,n$$

$$\theta_i^R, \theta_i^p, \theta_i^r \in \{0,1\}, \quad i = 1,2,...,n$$

In the above model we have used the symbol θ to represent the vector with components $\theta_i^R, \theta_i^p, \theta_i^r$, $i = 1,2,...,n$. Note that the expression to be minimized is the total delivery cost for all objects to clients in all regions. However, dividing this expression by P gives the cost for delivery to an individual region, whose clients collectively pay for $\frac{1}{P}$ of the remote delivery cost, since the regional client populations are statistically the same. As intuition suggests, the cache content that minimizes total delivery cost is the same as the content that minimizes an individual region's cost in the homogeneous system. Note that if $\beta = 0$, the model computes the proxy cache content that minimizes the use of remote server bandwith.

As explained above, the required remote server bandwidth (C_{remote}) and proxy server bandwidth ($C_{regional}$) are computed using relatively simple analytic expressions involving the total client request rate for each object (λ_i), the configuration parameters for the partitioned dynamic skyscraper delivery, and the object segment sets that are stored at the proxy. Similarly, the required storage at each proxy server ($D_{regional}$) is computed by summing over the object segments that are cached at the proxy servers. For the details of these equations, the reader is referred to [12].

Given the model input parameter values, the minimum cost cache content (i.e., the θ_i values) is computed through solution of a mixed integer linear program (MIP) [18].

Previous results of the model [12] showed that in homogeneous systems, it is often more cost-effective to cache the initial segments of many objects, rather than the complete data for fewer objects. The results also showed that the minimum cost cache content depends on key system parameters, including β and the relative constraints of disk bandwidth and storage capacity at the proxy servers. In particular, the results showed that as β increases or when the regional server bandwidth constraint is active for the minimum cost cache content, the proxy servers should cache segments of less popular objects.

Table 2: **New Parameters for the Request Rate and Server Heterogeneity Optimization Model**

Input	**Parameter Definition**
f_d	fraction of the total requests that are from clients belonging to the distinct region
Output	**Parameter Definition**
$C_{remote}^d, C_{remote}^{nd}$	the component of the remote server "cost" (as measured in numbers of channels) apportioned to the distinct proxy server, and to each other proxy server, respectively
$\theta_i^{y,z}$ $y,z \in \{R, p, r\}$	equals 1 if object *i* is cached at distinct and non-distinct proxy servers according to the superscripts *y* and *z*, respectively (*R* – the object is not cached regionally; *p* – only part (the leading segment set) is cached regionally; *r* – the object is fully cached at the respective regional server); equals 0 otherwise.

C. Optimization Models for Heterogeneous Systems

We develop two new optimization models that permit study of heterogeneous systems in which regions may have differing client request rates, proxy server capabilities, or object selection frequencies. The goal is to create models that are analytically tractable and can be used to derive useful insights and design principles for cache content in such heterogeneous systems.

The model in Section C.1 focuses on the impact of heterogeneous regional client request rates and server capabilities. The model in Section C.2 is designed to study the impact of heterogeneous object selection frequencies.

In each heterogeneous system model we assume that, for the case of competitive proxy service providers, each proxy server that does not cache a given segment set shares equally in the cost of the remote multicasts of those segments, even if the multicasts are used by different numbers of clients in each region. This pricing policy is motivated by the fact that a competitive proxy service provider may not wish to report its client request rate characteristics.

C.1 Heterogeneous Client Request Rates and Server Capabilities

To create a tractable yet useful model, one proxy server is assumed to have a higher or lower client request rate, and possibly also different bandwidth and storage capacity, than all other proxy servers which have the same request rate and server capabilities.

Letting *d* denote the distinct server that has higher or lower client request rate, and *nd* denote any of the other (non-distinct) proxy servers, Table 2 defines the new input and output parameters for this heterogeneous model. In addition to these new parameters, the input and output parameters for the proxy servers ($N_{channels}$, $N_{segments}$, $C_{regional}$, and $D_{regional}$) each have a superscript (*d* or *nd*) to denote the type of server (distinct or non-distinct) that the parameter applies to. As before, the key outputs of the model are the θ_i parameters that specify whether each object *i* is fully or partially cached at each type of regional proxy server. Note that there are nine such parameters for each object, due to all possible pairs of superscripts on the θ_i values.

Object allocations that minimize total delivery cost are obtained by solving the following optimization problem:

$$\min_{\theta} \quad C_{remote}(\theta) + \beta(C^{d}_{regional}(\theta) + (P-1)C^{nd}_{regional}(\theta))$$

subject to $C^{d}_{regional}(\theta) \le N^{d}_{channels}$

$$C^{nd}_{regional}(\theta) \le N^{nd}_{channels}$$

$$D^{d}_{regional}(\theta) \le N^{d}_{segments}$$

$$D^{nd}_{regional}(\theta) \le N^{nd}_{segments}$$

$$\sum_{y,z \in \{R,p,r\}} \theta_i^{y,z} = 1, \qquad i = 1,2,...,n$$

$$\theta_i^{y,z} \in \{0,1\}, \quad y,z \in \{R,p,r\}, \quad i = 1,2,...,n$$

Note that we have used the notation θ to represent the vector whose components are $\theta^{y,z}$, $y,z \in \{R,p,r\}$, $i = 1,2,...,n$. As in the homogeneous system model, the storage requirements for each type of proxy server are computed by summing over the segments stored at that type of server. The calculations of required bandwidth for the remote server and each type of proxy server use the same approach as in the homogeneous model, but are more complex due to the possibility that each type of proxy server may optimally cache different segments. The detailed equations are provided in [13].

We use the term "socially optimal" to denote the proxy server cache content that minimizes total delivery cost, as in the above optimization model.

In heterogeneous systems that have asymmetric client workloads or diverse server capabilities, the "socially optimal" cache content may differ from the "individually optimal" cache content that would result if a particular (competitive) regional service provider attempted to minimize its own delivery cost. To assess whether these two types of solutions differ for a given system specified by the model inputs, we also derive individually optimal cache content for the distinct server, under a fixed set of cache contents (e.g., the socially optimal allocations), for the non-distinct proxy servers. Similarly, we derive the individually optimal cache content for the non-distinct proxy servers, under a fixed cache content for the distinct proxy server.

The following optimization problem minimizes the delivery cost of the distinct regional server, with the superscript *O* in $\theta^{y,O}$ denoting the fixed allocations assumed for the non-distinct regions:

$$\min_{\theta} \quad C^{d}_{remote}(\theta) + \beta C^{d}_{regional}(\theta)$$

subject to $C^{d}_{regional}(\theta) \le N^{d}_{channels}$

$$D^{d}_{regional}(\theta) \le N^{d}_{segments}$$

$$\sum_{y \in \{R,p,r\}} \theta_i^{y,O} = 1, \qquad i = 1,2,...,n$$

$$\theta_i^{y,O} \in \{0,1\}, \quad y \in \{R,p,r\}, \quad i = 1,2,...,n$$

The delivery cost expression that is minimized in the above model includes only the cost of remote server multicasts that is apportioned to the distinct region. This cost is the sum of (a) $\frac{1}{P}$ of the channels required to multicast segments that are not cached by any of the servers, and (b) all of the bandwidth required to multicast segments that are not stored at the distinct proxy server but are stored at the other proxy servers. The detailed equations are given in [13].

C.2 Heterogeneous Object Selection Frequencies

In this model, the objects and the proxy servers are each partitioned into *G* equal-sized groups, and each group of proxy servers has a *preference* (i.e., a larger fraction of the regions' client requests) for a distinct group of objects. Each proxy server has the same request rate for each of its (*G*-1) non-preferred groups of objects. Also, the *relative* selection frequencies of the objects *within a group* are the same for all groups and for all proxy servers.

The new model notation for the system with the above heterogeneous object selection frequencies is given in Table 3. Note that although each proxy server may optimally cache different object segments, due to the symmetry in the regional client workloads, each proxy server will cache the same segments from its respective preferred and non-preferred groups. This greatly simplifies the model notation, and leads to an analytically tractable model that again has nine cache placement variables (θ_i) per object.

The symmetry in the regional client workloads also implies that the socially optimal proxy cache content is also the individually optimal. This optimal cache content is computed by solving the following optimization problem:

$$\min_{\theta} \quad C_{remote}(\theta) + P\beta C_{regional}(\theta)$$

$$\text{subject to} \quad C_{regional}(\theta) \le N_{channels}$$

$$D_{regional}(\theta) \le N_{segments}$$

$$\sum_{y,z \in \{R,p,r\}} \theta_i^{y,z} = 1, \qquad i = 1,2,...,n$$

$$\theta_i^{y,z} \in \{0,1\}, \quad y,z \in \{R,p,r\}, \quad i = 1,2,...,n$$

The server bandwidths and storage requirements are computed using the same approach as in the previous models. The complete set of equations is given in [13].

C.3. Solution of the Heterogeneous Optimization Models

Although the detailed equations for the heterogeneous models defined above are somewhat cumbersome [13], both the objective functions and constraints in each model are *linear* functions of the binary variables θ. This was a key element in our design of the models; as these problems are tractable mixed integer programs (MIPs) [18].

For the purposes of obtaining solutions to the models, we formulated all the optimization problems in the GAMS modeling language [3] and solved them using a combination of the XPRESS [10] and CPLEX [8] solvers. Both of these codes use a linear programming [9] based branch and bound solution strategy.

Table 3: **New Parameters for the Object Selection Frequency Heterogeneity Optimization Model**

Input	**Parameter Definition**
f_p	fraction of the total requests from a region that are for its preferred group of objects
G	number of groups of objects
Output	**Parameter Definition**
$\theta_i^{y,z}$ $y,z \in \{R,p,r\}$	equals 1 if object *i* is cached at each proxy server for which the object's group is the region's preferred group, and at each proxy server for which the object's group is not the preferred group, according to the superscripts *y* and *z*, respectively (*R* – the object is not cached regionally; *p* – only the leading segment set is cached at a preferring/non-preferring regional server; *r* – the object is fully cached at a preferring/non-preferring regional server); equals 0 otherwise.

D. Insights and Design Principles

This section presents results of the optimization models for three types of heterogeneous systems:

1) systems with *P*-1 homogeneous servers and one server with significantly lighter client request rate,

2) systems with *P*-1 homogeneous servers and one server with significantly heavier request rate, and

3) systems with four groups of regional servers, each group having a higher fraction of requests for its preferred group of objects.

In each case we are interested in understanding the impact of the heterogeneity on the optimal proxy cache content. For the first two types of systems we are also interested in the extent to which there is a divergence between the proxy cache content that minimizes total delivery cost and the content that minimizes cost for the clients local to each type of server. These individually optimal caching strategies are important if competitive organizations are providing the proxy servers. The key insights from the results of the model are summarized in section D.4.

In each experiment discussed below, the skyscraper configuration parameters are set to *K*=12, *W*=64, *k*=5, and $s_j = (1,2,2,4,4,...)$. In this system, the leading segment set is the first five segments of an object, which corresponds to the first 6.8% of the data file. Request arrival rates are expressed in terms of the total number of client requests that arrive during a unit-segment delivery time. The storage and bandwidth capabilities for a regional server are stated in terms of multiples of a "baseline capability" that is assumed to be 2436 segments and 192 channels[2].

Object selection frequencies (overall, or within a group of objects in the model defined in section C.2) are assumed to have the Zipf(0) distribution.

In each graph below, the height of the black bar for each object, in order of decreasing object popularity, indicates the percentage of the object are stored at a given proxy server in a given minimum cost solution.

D.1 One Proxy Server with Light Request-Rate

The system considered here has ten regional servers and a total client request rate (λ) equal to 100 requests per unit-segment multicast period. One proxy server's client request rate is equal to 1.25; the other nine proxy servers each have identical client request rate nearly equal to 11.

[2] This baseline capability is derived by assuming each object is of size 1.35 gigabytes and delivery rate equal to 1.5 million bits per second, and the server has four commodity disks that each have capacity of 4.35 gigabytes.

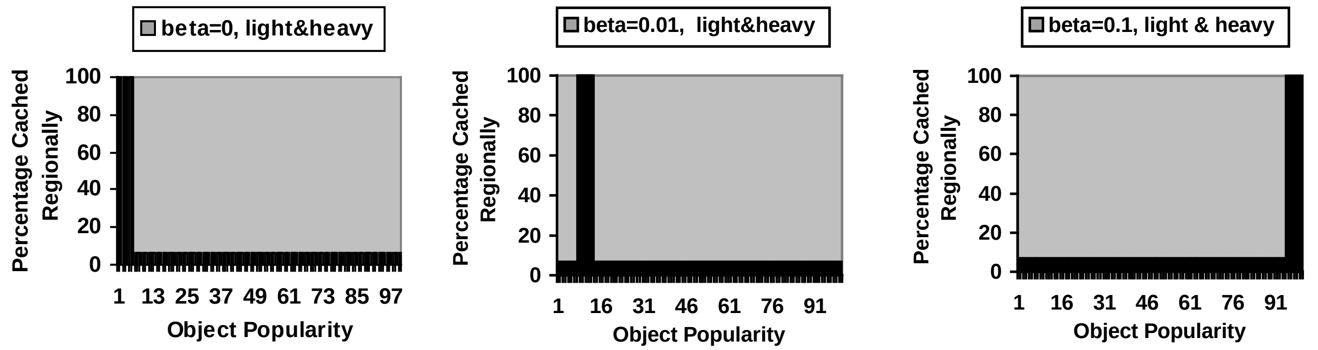

Figure 1: Socially Optimal Caching for Baseline System with One Light Region

Figure 1 presents the proxy cache content that minimizes total delivery cost for this heterogeneous system when the relative cost of regional and remote server channels, β, is equal to 0, 0.01, or 0.1. As noted in section C, the optimal proxy cache configuration is sensitive to the value of β. Note that, for this heterogeneous system,

- The cache content that minimizes total delivery cost is the same for both types of proxy servers.
- We have verified that the cache content that minimizes the delivery cost to clients in either type of proxy server, while fixing the content of the other type of server as given in figure 1, yields (nearly) the same cache content as the content that minimizes the total delivery cost. Thus, in this heterogeneous system, competitive proxy service providers can individually optimize the data stored at their respective servers and arrive at the socially optimal cache configuration.
- We have verified that this cache configuration is nearly identical to the optimal cache content for a system with ten proxy servers that each have request rate equal to 10.

Thus, when all proxy servers have the same bandwidths and storage capacity, the single proxy server with light request rate does not appreciably influence the optimal content of the heavy request rate regions, and the individually optimal cache configurations are the same as the socially optimal configurations.

Figure 2 gives the light and heavy proxy server cache configurations that minimize total delivery cost when the light server has 25% of the bandwidth and storage at the proxy servers with heavy client request rate. Results are shown for β equal to zero; the results for β equal to 0.01 are very similar. In this case, the light request-rate server only stores the initial 6.8% of the data for about 30 of most popular objects, and the heavy request rate regions should fully cache the least popular objects, rather than the most popular objects that they store in Figure 1. This shows a very strong influence of the light server on the caching at heavy servers, perhaps stronger than one might anticipate. The explanation for this socially optimal heavy server caching strategy is that the central server must multicast the trailing segment set of the most popular objects fairly frequently for the light server, so the greatest global advantage is for the heavy servers to share the cost of those multicasts and instead cache the segments that the light server needs very infrequently.

If we minimize the delivery cost for clients of the heavy regional servers while keeping the cached segments at the light server as in Figure 2, we find that the heavy servers will fully cache the most popular objects as they do in the baseline heterogeneous system.[3] Thus, the individually optimal cached segments for the proxy servers with heavy client load differs dramatically from the socially optimal.

We also investigated cases where the proxy server with light client load has increased storage and bandwidth equal to twice the storage and bandwidth at the proxy servers with heavy client load. In some cases, for example when beta equals 0.1, to minimize total delivery cost the light server caches exactly the same segments as the heavy servers; the extra storage and bandwidth capabilities are not used. However, the individually optimal solution for the light server, with the cached segments in the heavy server fixed as they are for minimizing total delivery cost, shows that a competitive light server would store additional remote data.

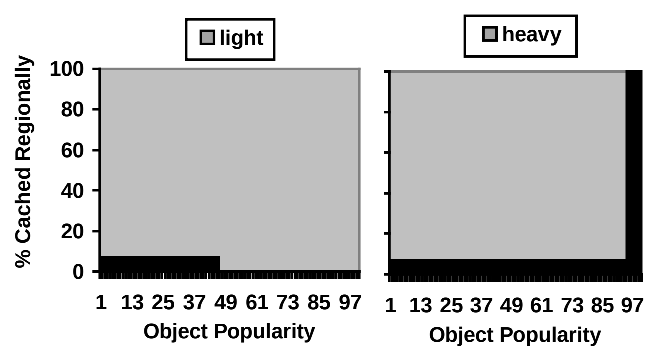

Figure 2: Socially Optimal Caching for System Containing One Light Server with Reduced Storage ($\beta = 0$)

[3] If we minimize the delivery cost for the light server while fixing the individually optimal cached segments at the heavy servers, the light server caching strategy remains as in Fig. 2.

D.2 One Proxy Server with Heavy Request-Rate

The first system considered here has ten proxy servers and total client request rate equal to 20 requests per unit-segment multicast period. One proxy server's client request rate is equal to 10. The other nine proxy servers have equal client request rate of about 1 request per unit segment multicast period. All servers have the baseline bandwidth and storage capability.

The results for this heterogeneous system with β equal to 0.1 are given in Figure 3. To minimize total delivery cost (Figure 3a), both types of proxy servers store the same data. Furthermore, the caching strategy is similar to the optimal cache content in a homogeneous system with similar total request rate, but the particular files that are fully cached are a "compromise" between the (more popular) files that should be fully cached if every server had request rate equal to 1, and the (less popular) files that should be fully cached if every server had request rate equal to 2.

Minimizing the light server delivery cost while fixing the cached segments in the heavy server as in the results for optimizing total delivery cost, and then minimizing each light server's delivery cost under the fixed individually-optimal caching strategy for the heavy server, results in the individually optimal caching strategy given in Figure 3b, which again differs from the socially optimal caching strategy in Figure 3a. Furthermore, if the light server cache configurations are fixed at these individually optimal values, the heavy server's individually optimal strategy is to cache the same segments as the light server. In this case, it's possible that continuing to competitively optimize individual regional server delivery costs may lead to an unstable solution. One of the values of the optimization model is to discover such situations in which one might want to devise new pricing strategies.

The second heterogeneous system with one heavy request-rate region has: (1) an overall client request rate per unit-segment multicast period equal to 100, (2) one region's client request rate equal to 20, and (3) each of the other nine regions' client request rate approximately equal to 9. The light servers have the baseline bandwidth and storage capacity, whereas the heavy server has twice the baseline server capabilities.

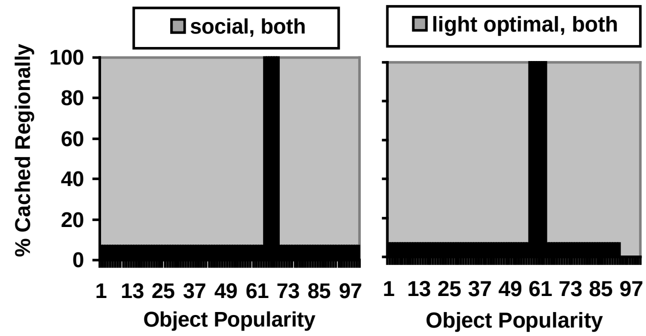

Figure 3: Optimal Caching for System Containing One Relatively Very Heavy-Rate Server ($\lambda = 20$, $\lambda_{heavy} = 10$, $\lambda_{light} = 1$, $\beta = 0.1$)

Results for β equal to 0 and 0.1 are presented in Figure 4. In the caching strategy that minimizes total delivery cost for the heterogeneous system, when $\beta = 0$, each server caches what is individually optimal. However, when $\beta > 0$, the heavy server caches exactly what the light servers cache; its extra storage and bandwidth are not helpful in reducing cost. However, the individually optimal solution for the proxy server with a heavy client request rate, fixing light server caching as in the system that minimizes total delivery cost, is given in Figure 4c. This result shows that the delivery cost for the heavy-rate region is minimized when additional segments are cached.

D.3 Heterogeneous Object Popularities

We consider here systems with twelve regional servers, partitioned into four groups of three servers, and two hundred objects, also partitioned into four groups, of fifty objects each. Each group of proxy servers has a different preferred group of objects. All non-preferred groups of objects are selected equally often. Selection frequency within each group (preferred or not) is modeled by a Zipf(0) distribution. The

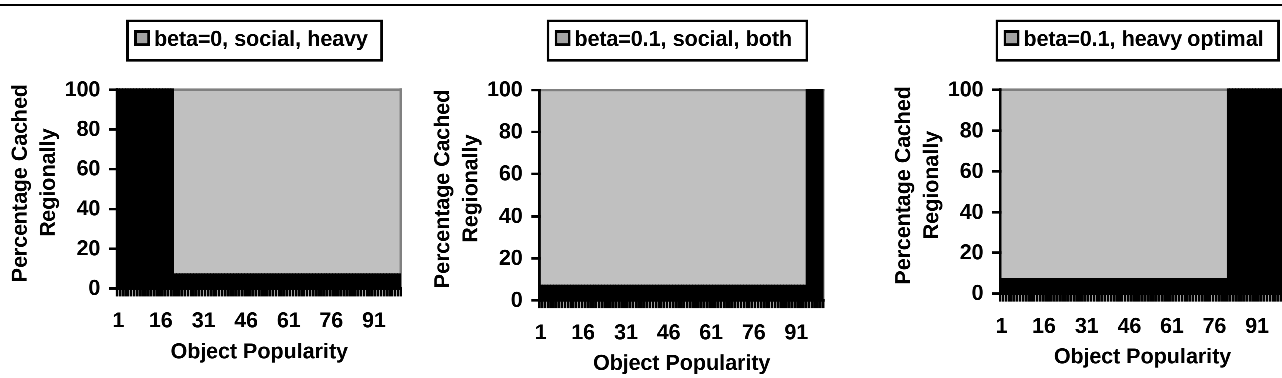

Figure 4: Optimal Caching for System Containing One Relatively Heavy-Rate Server ($\lambda = 100$, $\lambda_{heavy} = 20$, $\lambda_{light} = 10$)

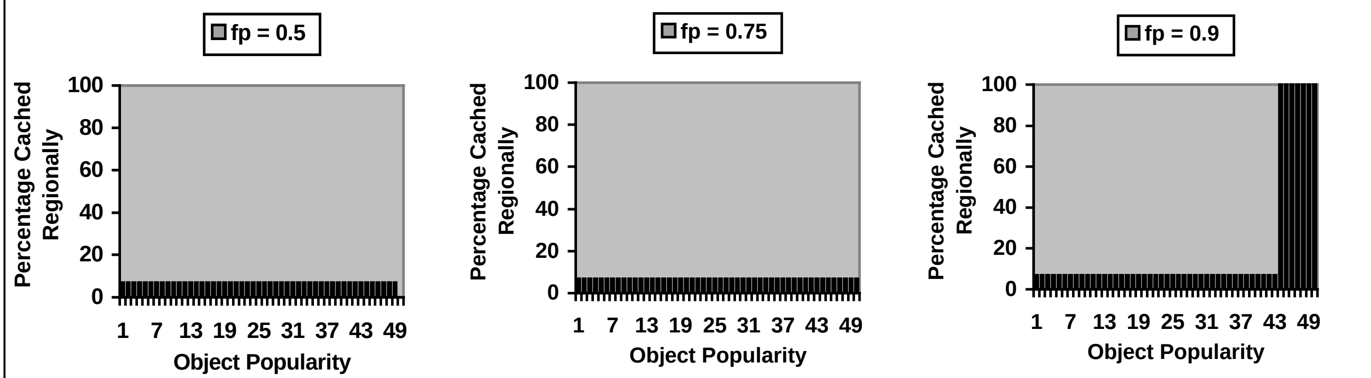

Figure 5: Optimal Caching for a Preferred Group (baseline capabilities)

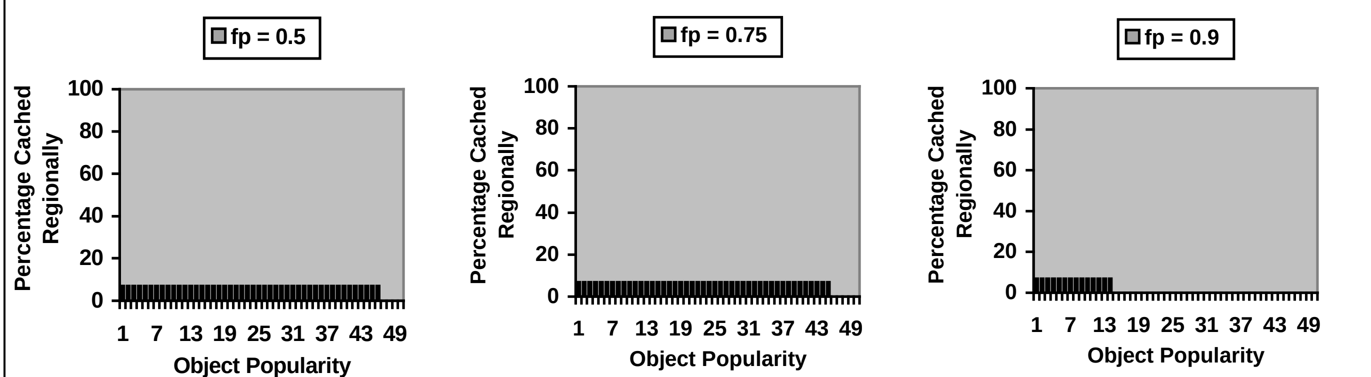

Figure 6: Optimal Caching for a Non-Preferred Group (baseline capabilities)

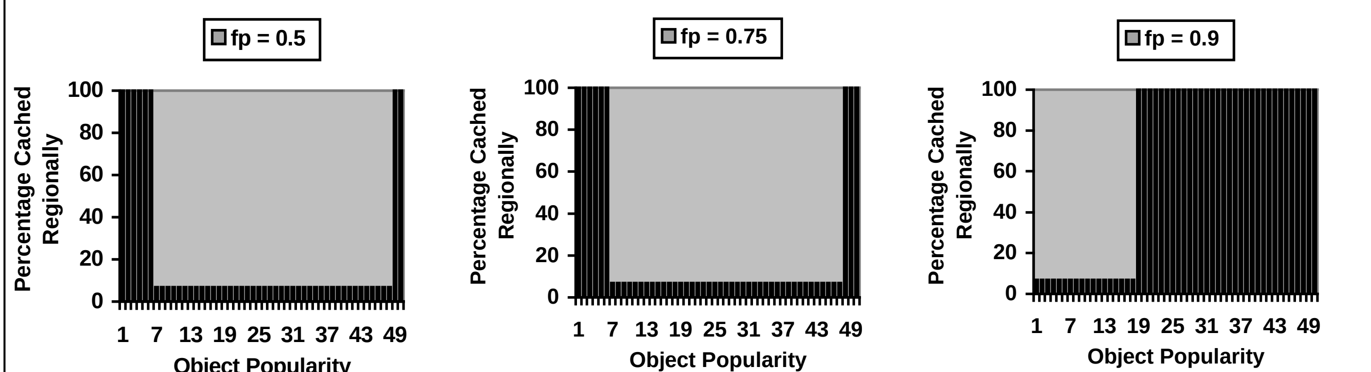

Figure 7: Optimal Caching for a Preferred Group (3 times baseline capabilities)

total client request rate per unit-segment multicast period, $\lambda' = 100$.

Figures 5 and 6 show how the preferred and non-preferred groups of objects should be cached to minimize (total and individual) delivery cost, for the case in which each server has the baseline bandwidth and storage capability and $\beta = 0$. When 50% or even 75% of the client requests are directed towards the preferred group (i.e., $f_p = 0.5$ or $f_p = 0.75$), the proxy servers should store nearly identical data for both the preferred and non-preferred objects. This provides further illustration of the tendency for optimal configurations to be ones in which the regional servers keep very similar cache contents. Furthermore, as can be seen in the results for $f_p = 0.9$, even when heterogeneity is so strong that the optimal cache content of proxy servers with differing preferred groups begins to diverge, this divergence occurs for the least popular objects within each group rather than for the most popular. This reflects the inefficiency that would result from having multicasts of relatively popular objects from the central server, if these same objects are cached at only some of the regional servers.

Figure 7 shows the optimal caching strategy for the objects in a proxy server's preferred group, if $\beta = 0$ and each server has three times the baseline bandwidth and storage capability. The corresponding caching strategy for objects within a non-preferred group (not shown), (1) is identical for the most popular objects, (2) diverges to a small extent for the least popular objects for $f_p = 0.5$ or $f_p = 0.75$ and (3) is greatly different when $f_p = 0.9$, but only for objects of lesser relative popularity, with cache allocations similar in nature to those in Figure 6.

Generally, common caching of intial segments appears to be a desirable compromise when different regions have different object selection frequencies. Furthermore, commonality in cache contents is seen to be most critical for relatively more popular objects. There is a greater tendency for both whole object caching, and no caching at all, for the least popular objects in heterogeneous systems.

D.4 Summary of Insights and Design Principles

1. It is often optimal to cache the initial segments of many objects rather than all segments of fewer objects. In the case of heterogeneous object selection frequencies, there is an increased tendency to cache initial segments rather than full objects, owing to the improved cost-sharing for remote delivery when all servers cache the same segments, and since the aggregate selection frequency distribution is less skewed.

2. Minimizing the total delivery cost often influences all servers to store the same segments. For example, a single region with light request rate relative to all other regions should often cache the set of segments that is optimal for the heavy request-rate regions. There are also cases where a single region with a heavy request rate may influence all other light-rate regions to cache a different set of segments than they would otherwise cache.

3. When minimizing the total cost of delivery, a (group of) regional server(s) with distinct behavior can sometimes influence the set of segments cached by the rest of the servers in unexpected ways. For example, under conditions where a set of homogeneous servers would fully cache the *most popular* objects and partially store the rest of the objects, adding a regional server that has a significantly lighter request rate might influence the rest of the servers to fully cache the *least popular* objects and partially store all other objects.

4. When minimizing the delivery cost for a given proxy server, the resulting optimal set of segments to be cached at that server may be quite different than the "socially optimal" set that would minimize overall cost of delivery. Thus, it may be in the remote provider's best interest to devise (artificial) pricing formulas to encourage the socially optimal caching strategies.

E. CONCLUSIONS

Caching data close to the requesting users can greatly reduce the network bandwidth required to achieve wide-area sharing of large, popular data objects. For data such as video, effective policies for determining what to cache are crucial, as the storage and server bandwidth requirements of such objects do not permit indiscriminate caching of all requested objects. The issue of what to retain in cache is complicated by multiple constraints (storage capacity and bandwidth), the multicast delivery of partial data streams.

In this paper we have addressed this issue in the context of a shared central server and multiple regional (proxy) servers, and multicast delivery of segmented data streams. New analytic models for determining optimal cache configurations have been developed, which permit investigation of more realistic (*i.e.*, heterogeneous) systems than in previous work. Heterogeneity is a crucial issue for proxy caching, as it implies a fundamental conflict between the objectives of caching the data most useful locally, and of maximizing commonality in cache contents (and thus cache misses) across regional servers so as to achieve the greatest possible sharing of multicasts from a remote server. Results of the new models show that it is often more cost-effective to cache just the initial segments of objects, to cache similar sets of objects at all the regional servers, and to rely on remote multicast delivery for some portions of hot objects while caching locally corresponding portions of cooler objects.

In a dynamic environment with changing object popularities, in addition to knowing what might be optimal to retain in cache for given workload parameters, cache replacement policies are needed that (approximately) achieve such configurations. On-going research includes developing such cache replacement policies, as well as investigating optimal cache content for (1) multicast delivery techniques that permit more flexible partitioning of the data between the proxy and the remote server and (2) layered continuous media data.

References

[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "A Permutation Based Pyramid Broadcasting Scheme for Video-on-Demand Systems", *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Hiroshima, Japan, June 1996.

[2] C. C. Bisdikian and B. V. Patel, "Cost-Based Program Allocation for Distributed Multimedia-on-Demand Systems", *IEEE MultiMedia 3*, 3 (Fall 1996), pp. 62-72.

[3] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*, The Scientific Press, South San Francisco, CA, 1988.

[4] D. W. Brubeck and L. W. Rowe, "Hierarchical Storage Management in a Distributed VOD System", *IEEE MultiMedia 3*, 3 (Fall 1996), pp. 37-47.

[5] S. W. Carter and D. D. E. Long, "Optimizing Patching Performance", *Proceedings of the 6th International Conference on Computer Communications and Networks (ICCCN'97)*, Las Vegas, Nevada, September 1997, pp. 200-207.

[6] P. Cao and S. Irani, "Cost-Aware WWW Proxy Caching Algorithms", *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS)*, Monterey, CA, December 1997, pp. 193-206.

[7] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrell, "A Hierarchical Internet Object Cache", *Proceedings of the 1996 USENIX Technical Conference*, January 1996.

[8] CPLEX Version 6.0. ILOG CPLEX Division, Incline Village, Nevada (http://www.cplex.com/).

[9] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

[10] *XPRESS-MP User Guide*, Dash Associates, Blisworth House, Blisworth, Northants, UK, 1997 (http://www.dashopt.com/).

[11] D. L. Eager and M. K. Vernon, "Dynamic Skyscraper Broadcasts for Video-on-Demand", *Proceedings of the 4th International Workshop on Multimedia Information Systems (MIS '98)*, Istanbul, Turkey, September 1998, pp. 18-32.

[12] D. L. Eager, M. C. Ferris, and M. K. Vernon, "Optimized Regional Caching for On-Demand Data Delivery", *Proceedings of the IS&T/SPIE Conference on Multimedia Computing and Networking 1999 (MMCN '99)*, San Jose, CA, January 1999, pp. 301-316.

[13] D. L. Eager, M. C. Ferris, and M. K. Vernon, "Models for Optimized Caching in Systems with Heterogeneous Client Populations", Technical Report 1402, University of Wisconsin-Madison, Aug 1999.

[14] D. L. Eager, M. K. Vernon, and J. Zahorjan, "Minimizing Bandwidth Requirements for On-Demand Data Delivery", to appear in the *Proc. Of the 5th International Workshopon Multimedia Information Systems (MIS '99)*, Indian Wells, CA, October 1999.

[15] L. Gao, J. Kurose and D. Towsley, "Efficient Schemes for Broadcasting Popular Videos", *Proceedings of the 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98)*, Cambridge, UK, July 1998.

[16] K. A. Hua and S. Sheu, "Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems", *Proceedings of the ACM SIGCOMM'97 Conference*, Cannes, France, September 1997, pp. 89-100.

[17] K. A. Hua, Y. Cai, and S. Sheu, "Patching: A Multicast Technique for True Video-on-Demand Services", *Proceedings of the 6th ACM International Conference (ACM MULTIMEDIA '98)*, Bristol, U.K., September 1998, pp 191-200.

[18] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York, NY, 1988.

[19] J.-P. Nussbaumer, B. V. Patel, F. Schaffa, and J. P. G. Sterbenz, "Networking Requirements for Interactive Video on Demand", *IEEE Journal on Selected Areas in Communications 13*, 5 (June 1995), pp. 779-787.

[20] S. Sen, J. Rexford, and D. Towsley, "Proxy Prefix Caching for Multimedia Streams", *Proceedings of IEEE Infocom '99*, New York, NY, March 1999.

[21] R. Tewari, H. M. Vin, A. Dan, and D. Sitaram, "Resource-based Caching for Web Servers", *Proceedings of the IS&T/SPIE Conference on Multimedia Computing and Networking 1998 (MMCN '98)*, San Jose, California, January 1998.

[22] S. Viswanathan and T. Imielinski, "Pyramid Broadcasting for Video-on-Demand Service", *Proceedings of the IS&T/SPIE Multimedia Computing and Networking Conference 1995 (MMCN'95)*, San Jose, CA, February 1995, pp. 66-77.

[23] S. Viswanathan and T. Imielinski, "Metropolitan Area Video-on-Demand Service using Pyramid Broadcasting", *Multimedia Systems 4*, 4 (August 1996), pp. 197-208.