Optimized Caching in Systems with Heterogeneous Client Populations*

Derek L. Eager **Michael C. Ferris** **Mary K. Vernon**
University of Saskatchewan University of Wisconsin - Madison
Saskatoon, SK Canada S7N 5A9 Madison, WI 53706-1685

Abstract

An important question in providing on-demand access to large widely shared data files, such as popular video files, is how to effectively use regional (proxy) servers that can 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 highly efficient analytic models for determining optimal proxy cache content in such environments. Specifically, the new models apply to heterogeneous systems in which the proxy servers have different client workloads and server capabilities. Results from the models provide general insights into caching strategies for such systems, and suggest that it may be useful to employ efficient cost models in actual systems to determine what content should be cached in response to the measured client workload.

1 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. This paper addresses the question of which large popular widely shared data should be stored at the regional/proxy servers. The context is of a system containing multiple heterogeneous proxy servers, one or more shared remote servers (i.e., the information sources, which are distinct from the proxy servers), and *multicast delivery* to conserve server and network bandwidth. We assume the system uses the recently proposed partitioned dynamic skyscraper delivery technique [11] reviewed in Section 2. As explained in Section 2.1, our results are also applicable to other on-demand multicast delivery techniques. 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. However, the models are highly efficient, and typical solution time is within a few seconds. Thus, the models serve as "prototypes" for cost models that could be employed in actual systems to compute the optimal cache content for a given measured client workload.

Most prior work on World-Wide Web caching as well as distributed video-on-demand (VOD) architectures has focused on determining on which server(s) each entire file should be allocated, so as to optimize system cost/performance [21, 2, 4, 8, 6]. Other related work has concerned strategies for dynamically caching intervals of data from continuous media Web files, so as to satisfy one or more requests that arrive close in time to a previous request [24]. This prior work on whole file placement and interval caching has not considered the impact of *shared delivery of popular files* that is enabled by multicast delivery techniques. In particular, there is a new trade-off between caching the files that are requested most frequently and caching less popular files that have less cost sharing when delivered from the remote server. Furthermore, for segmented multicast delivery techniques such as partitioned dynamic skyscraper, the earlier segments of a file 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 later segments of the file. This leads to another key cost trade-off between caching the entire data for a particular popular file or caching the initial segments of many more files.[1]

Heterogeneity in proxy workloads and server capabilities is an important factor because it occurs in practical systems and because it introduces a new tension between (a) tailoring the data stored at each proxy according to the

---

* This research was partially supported by the Natural Sciences and Engineering Research Council of Canada under Grant OGP-0000264, by the Air Force Office of Scientific Research under Grant F49620-98-1-0417, and by the National Science Foundation under Grant CCR 9975044. To appear in *Performance Evaluation,* September 2000.

[1] Recent work has established that caching just the initial segments of objects has a number of other advantages, including hiding the latency of communication with a remote server, and facilitating workahead smoothing of variable-bit-rate video [23].

local client workload, and (b) maximizing uniformity in proxy cache contents so as to achieve the greatest possible sharing of remove server multicasts of uncached items. 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 and apply models that provide insight into how this tension is resolved for various kinds of heterogeneity.

We consider the above trade-offs in the context of determining which data should be cached on the disks of a regional proxy server. We consider that the popular files on a given multimedia server, such as the popular lectures on a distance education server or the popular shows on an entertainment server, might be expected to remain in high demand for perhaps as long as a good fraction of a day, or longer. Requests for low popularity content on the server will be interspersed with requests for the popular content. Due to disk bandwidth considerations at the proxy server, we believe that it makes sense to semi-statically cache the popular files (i.e., over periods of relatively stable measured access rates), rather than dynamically caching the files in response to individual client requests. The question addressed in this paper is, for a given set of files with specified request rates, which files or portions of the files should be semi-statically cached in order to minimize the delivery cost for the files?

Models that include client cost-sharing, as well as server bandwidth requirements for the new multicast delivery techniques, are needed to evaluate the semi-static cache design trade-offs. A previous optimization model [11] reviewed in Section 2.2 determines what data should be stored at *homogeneous* proxy servers. The model computes the cache content that minimizes a specified delivery cost objective, 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 file. We modify 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.

A key challenge is to create models that are efficient to solve and that allow us to gain the desired insights for heterogeneous systems. We have developed two such models, which are described in Section 3. The models are applied in Section 4 to study proxy server data caching strategies and the impact of heterogeneity. The insights and design principles derived from the results include:

- Even in systems with quite strong heterogeneous features, it is often more cost effective to store *the same data at all of the proxy servers*, rather than to closely tailor cache content at each proxy according to the local client preferences and server characteristics.
- It is often more cost-effective to store just initial segments rather than entire files at the proxy servers. Heterogeneity increases this tendency as compared with the homogeneous system.
- When minimum total delivery cost is the objective, a regional proxy server with a distinct client workload can sometimes influence the data cached by the other regional proxy servers in unexpected ways.
- When minimum delivery cost for the clients of a given proxy server is the objective, 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.

Cases where the latter two observations apply motivate the use of efficient cost models to guide cache content in actual systems. Section 5 provides the conclusions of this paper and suggests fruitful topics for future research.

2 Background

Section 2.1 reviews segmented multicast delivery techniques and the partitioned dynamic skyscraper delivery technique that is assumed in the delivery cost models developed in this paper. Section 2.2 reviews the previous cost model for determining optimal cache content in partitioned dynamic skyscraper systems with homogeneous proxy servers. The new models that will be developed in Section 3 can be applied to homogeneous systems as a special case of the input parameters; thus the detailed equations of the previous homogeneous system model are not repeated in this paper.

Throughout the remainder of the paper, a (proxy or remote) server is assumed to have enough disk bandwidth, memory bandwidth, and network i/o bandwidth to support a given number of simultaneous data streams at a specified file delivery rate. For simplicity, all files are assumed to have the same delivery rate, although the caching models can easily be extended to handle multiple file streaming rates. The fixed-rate streams will be referred to as "channels" throughout the paper, and server bandwidth will be measured in units of the (fixed-rate) channels that the server can support.

2.1 Segmented Multicast Delivery

The conventional approach to on-demand delivery of video and other continuous media files is to allocate a new data stream (or channel) for each client request. Recently proposed *segmented multicast* delivery techniques [26, 25, 1, 18, 16, 10, 11] achieve significant bandwidth savings by dividing each file into increasing-sized segments (s_1, s_2, s_3, …, s_K). The server then employs a multicast transmission schedule in which the smaller segments are multicast more frequently than the larger segments, such as the schedule illustrated in Figure 1. Each client receives one transmission of each segment, such as the gray shaded sequence of transmissions in Figure 1. By frequently multicasting the smallest first segment (s_1), the time that a client must wait to begin receiving the file is fairly small. Since the larger segments are multicast less frequently, clients must be prepared to receive those segments ahead of when they are needed for playback, batching together with other clients that may be at different play points. For example, a client who receives the striped s_1 transmission will need to receive segment s_2 concurrently, buffering the second segment until it is needed for playback. The structure of the transmission schedule ensures that, for any s_1 multicast a client might receive, the client can receive each other file segment before or at the time it is needed for playback. (The reader can verify this in Figure 1.) Multicasting the larger segments less frequently greatly reduces server bandwidth usage for high client request rates.

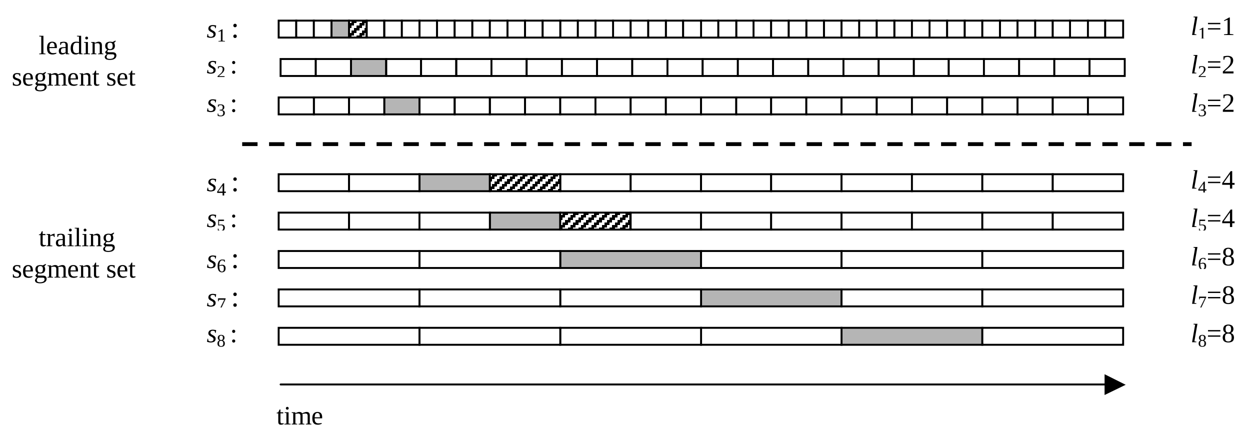


Figure 1: **Example Segmented Multicast Architecture – Partitioned Dynamic Skyscraper**
($K=8$, $W=8$, $k=3$, $w=2$)

Most of the segmentation-based delivery techniques employ static transmission schedules in which a fixed amount of server bandwidth (e.g., equal to K times the file streaming rate) is continuously assigned to multicasting the K segments of a specific popular file. An exception is the *partitioned dynamic skyscraper* technique [10, 11], which dynamically assigns server bandwidth to file segment transmissions in response to the incoming client requests.

The partitioned dynamic skyscraper technique partitions the file segments into two sets: a "leading segment set" (s_1,…,s_k) and a "trailing segment set" (s_{k+1},…,s_K). Server bandwidth equal to a *transmission cluster*, illustrated in Figure 1 by the shaded and striped transmission periods for each segment set, is scheduled as needed to deliver the respective file segment set in response to client requests. The leading segment transmission cluster uses bandwidth equal to k channels, and has a skewed duration w equal to the length of segment k (l_k) on each channel; the trailing segment transmission cluster uses bandwidth equal to $K–k$ channels and has a skewed duration $W = l_K$ on each channel. Scheduling of the leading and trailing clusters must be coordinated so that, for each client, jitter between playback of each set is avoided. Within this constraint, leading segment clusters are scheduled as early as possible and trailing segment clusters are scheduled as late as possible, so that the latter can be shared by the maximum number of future client requests. A new client request either receives the leading (trailing) file segments from a cluster that was scheduled for a previous request to the same file, or schedules a new leading (or trailing) segment transmission cluster. Note that when a cluster has finished delivering the first segment in the set, the bandwidth allocated to the cluster (equal to k or K-k times the file streaming rate) can be reallocated to a cluster of segment transmissions for a different file, depending on client requests. The models in this paper determine how much total server bandwidth must be allocated for leading segment transmission clusters and trailing segment transmission clusters, respectively, so that the appropriate cluster(s) can be allocated when each new client request arrives.

Dynamic scheduling can provide true immediate service to client requests [10,12], as well as segmented multicast delivery for less popular files, and in contexts where file popularity varies over time. Partitioned dynamic scheduling improves the efficiency of segmented multicast transmissions because the leading segment transmission clusters are smaller and are used more efficiently. Partitioned multicasts also improve proxy cache performance because, for each file, a regional proxy server may cache just the leading segment set, both sets of segments, or neither set.

Recent work [12] shows that the average server bandwidth required per file when using partitioned dynamic skyscraper multicasts for immediate on-demand file delivery grows asymptotically only logarithmically in the client request rate.

Other on-demand multicast delivery techniques, such as the technique known as stream tapping, grace patching, or controlled multicast [7,19,5,17], and the recently proposed hierarchical multicast stream merging and bandwidth skimming [12,13,14], also transmit initial portions of a popular continuous media file more frequently than later portions of the file. Systems that use these delivery techniques would have similar trade-offs with respect to optimal proxy cache content and similar optimal caching strategies (at the level of detail studied in Section 4) as for systems that use the partitioned dynamic skyscraper multicast technique.

2.2 Model for Computing Optimized Cache Content in Homogeneous Systems

A previous model [11] 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) files of equal size and delivery rate. For each file, the proxy servers can store either none, all, or just the leading segment set of the file. 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 the new models for systems with heterogeneous proxy server capabilities and workloads.

The homogeneous model uses simple analytic estimates of the time average of the number of remote server channels (C_{remote}) and proxy server channels ($C_{regional}$) used if each client is served immediately, under a given set of file request rates as a function of the file segment sets that are stored at the proxy. Results given in [11] show that these simple bandwidth estimates are very close to the *knee* of the curve of mean client waiting time versus the inverse of the number of available server channels, and that the knee of the curve is typically quite sharp. Since the system can provide immediate service to client requests, the average client waiting time is typically very close to zero near the (sharp) knee of the curve.

Table 1 gives the model input and output parameters. Note that the last four input parameters (k, K, l_j, and W) specify the configuration of the partitioned dynamic skyscraper system. Note also that the parameter β can include consideration of approximate network bandwidth costs as well as the server resource costs for each type of channel or stream. The key system constraints in computing the cache content that minimizes delivery cost 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 file i, whether the file should be fully cached, 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 previous homogeneous system model for optimal proxy cache content is defined as follows:

$$
\begin{array}{ll}
\underset{\theta}{\textbf{min}} & C_{remote}(\theta) + P\beta C_{regional}(\theta) \\
\textbf{subject to} & C_{regional}(\theta) \le N_{channels} \\
 & D_{regional}(\theta) \le N_{segments} \\
 & \theta_i^R, \theta_i^p, \theta_i^r \in \{0,1\}, \quad i = 1,2,...,n \\
 & \theta_i^R + \theta_i^p + \theta_i^r = 1, \quad i = 1,2,...,n
\end{array}
$$

In the above model, the symbol θ represents 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 files to clients in all regions. However, since the regional client populations are statistically the same, dividing the total delivery cost by P gives the cost for delivery to an individual region, whose clients collectively pay for $1/P$ of the remote delivery cost. Thus, 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 bandwidth.

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

Workload	**Parameter Definition**
n	number of files
λ_i	total request rate for file *i* (from all regions)
System	**Parameter Definition**
$N_{channels}$	maximum number of channels at each proxy server
$N_{segments}$	storage constraint at each proxy server (measured in units of the size of segment s_1)
P	number of proxy servers
β	the cost ratio of a proxy server channel and a remote server channel
k	number of segments in the leading segment set
K	total number of segments in the leading and trailing segment sets
l_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 the size of segment s_1)
θ_i^R	equals 1 if file *i* is stored only at the remote server; 0 otherwise
θ_i^p	equals 1 if only the leading segment set of file *i* is cached regionally; 0 otherwise
θ_i^r	equals 1 if the entire file *i* is cached regionally; 0 otherwise

Equations for computing C_{remote}, $C_{regional}$, and $D_{regional}$ for heterogeneous system models are given in the Appendix and discussed in the next section. For the details of the homogeneous model equations, which are special cases of the heterogeneous system equations, the reader is referred to [11].

3 Models for Optimized Cache Content in Heterogeneous Systems

In this section 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 file selection frequencies. The goal is to create efficient models that can be used to derive useful insights and design principles for cache content in such heterogeneous systems.

The model in Section 3.1 focuses on the impact of heterogeneous regional client request rates and server capabilities. The model in Section 3.2 is designed to study the impact of heterogeneous file selection frequencies.

In the calculations of individual proxy server cost, we assume that if the proxy server does not cache a given segment set, it shares equally in the cost of all remote multicasts of the segment set. This pricing policy is based on the notion that a proxy server that does not cache the segment set pays for a "subscription" for remote delivery of the file, with fee proportional to the average number of remote server channels or streams required to deliver the segment divided by the number of subscriptions. This policy simplifies the accounting at the remote server and may also simplify the communication between the proxy and the remote server. The policy provides a price break for a region with a higher request rate for a given segment set that isn't cached. However, the price break is very small, since server bandwidth usage grows only logarithmically in the request rate. It is also straightforward to modify the models so that the cost of each remote multicast is shared equally by the proxy servers that listen to the multicast, if the implementation effort for the more detailed accounting is deemed worthwhile.

3.1 Heterogeneous Client Request Rates and Server Capabilities

To create an efficient and 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 file *i* is fully or partially cached at each type of regional proxy server. Note that there are nine such parameters for each file, due to all possible pairs of superscripts on the θ_i values.

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**
$C^d_{remote}, C^{nd}_{remote}$	the component of the remote delivery "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 file *i* is cached at the distinct and non-distinct proxy servers according to the superscripts *y* and *z*, respectively (*R* – the file is not cached regionally; *p* – only part (the leading segment set) is cached regionally; *r* – the file is fully cached at the respective regional server); equals 0 otherwise.

File allocations that minimize *total delivery cost* are obtained by solving the following optimization model:

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

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 $\theta_i^{y,z}$, $y,z \in \{R,p,r\}$, $i = 1,2,...,n$. 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 remote server bandwidth and required bandwidth for each type of proxy server use the same approach as in the homogeneous model, but are more complex for two reasons. First, each type of proxy server may optimally cache different segment sets. Second, there are complex interactions with respect to how multicast transmission clusters are shared by clients from the different types of proxy servers. The detailed equations that capture these effects are provided in the Appendix.

We use the term "socially optimal" to denote the proxy server cache content that minimizes total delivery cost, as expressed in the above optimization model. In heterogeneous systems that have asymmetric client workloads or server capabilities, the "socially optimal" cache content may differ from the "individually optimal" cache content that is obtained if a particular (competitive) regional service provider attempts to minimize its own delivery cost. To assess whether these two types of solutions differ for a given system, we derive individually optimal cache content for the distinct proxy server, under a fixed cache content (such as the socially optimal content) at 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 proxy server, with the superscript O in $\theta_i^{y,O}$ 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) $1/P$ 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, and the analogous model for minimizing the delivery cost of the non-distinct proxy servers, are given in the Appendix.

3.2 Heterogeneous File Selection Frequencies

In this model, the files 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 files. Each proxy server has the same client request rate for each of its (G-1) non-preferred groups of files. Also, the *relative* selection frequencies of the files *within a group* are the same for all groups and for all proxy servers.

The new model notation for the system with these heterogeneous file selection frequencies is given in Table 3. Although each proxy server may optimally cache different file segments, the symmetry in the regional client workloads implies that each proxy server will cache the same segments from its respective preferred group of files. Similarly, each proxy server will cache the same segments from its respective non-preferred groups. This greatly simplifies the model notation, and leads to an efficient model that again has nine cache placement variables ($\theta_i^{y,z}$) per file.

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

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

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

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

$$\begin{aligned}
\textbf{subject to} \quad & C_{regional}(\theta) \le N_{channels} \\
& D_{regional}(\theta) \le N_{segments} \\
& \sum_{y,z \in \{R,p,r\}} \theta_i^{y,z} = 1, \quad i = 1,2,...,n \\
& \theta_i^{y,z} \in \{0,1\}, \quad y,z \in \{R, p, r\}, \quad i = 1,2,...,n
\end{aligned}$$

The server bandwidths and storage requirements are computed using the equations for heterogeneous file selection frequencies given in the Appendix.

3.3 Solution of the Heterogeneous Optimization Models

Although the detailed equations for the heterogeneous models defined above are somewhat complex, both the objective functions and constraints in each model are *linear* functions of the binary variables θ. This was a key goal in our design of the models, as these problems are tractable mixed integer programs (MIPs) [20].

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 linear programming [9] based branch and bound solution strategy. Details of several computational enhancements to the basic model are given in [15].

4 Results and Insights

Previous results in [11] showed that in homogeneous systems, it is often more cost-effective to cache the initial segments of many files, rather than the complete data for fewer files. The results also showed that the optimal 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 optimal cache content, the proxy servers should cache (segments of) less popular files.

This section presents results and insights for the following 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 of servers having a higher fraction of requests for a particular preferred group of files.

In each case we are interested in understanding the impact of the given type of heterogeneity on the optimal proxy cache design rules. For the first two types of systems we are also interested in the extent to which there is a divergence between the "socially optimal" proxy cache content that minimizes total delivery cost and the "individually optimal" content that minimizes the cost for the clients local to a given proxy server. The individually optimal caching strategies are important if competitive organizations are providing the proxy cache services. The key results in this section are summarized in Section 5.

In each experiment discussed below, the system parameters are as given in Table 4 unless otherwise stated. The default constraints on proxy storage ($N_{segments}$) and bandwidth ($N_{channels}$) in Table 4 will be called the "baseline system" capabilities. These baseline quantities were derived assuming the server has six disks, each with storage capacity equal to 3.2 whole files and bandwidth for transmitting 48 simultaneous streams. Note that $N_{segments}$ = 3654 implies storage for approximately 19% of the total data for the 100 files that clients can request. The default parameters of the skyscraper delivery architecture (K, k, l_j, and W) were selected to be representative of a delivery architecture that might be deployed in practice. This architecture has a sum of segment lengths equal to 189, and the sum of leading set segment sizes equal to 29; thus, the leading segment set is 29/189 or 15% of the file. For the experiments reported in this paper, the optimal caching strategy is very similar for other values of k, but k=7 achieves the lowest optimized delivery cost in almost every case. Thus we report the results for this value of k.

The transmission time for the first segment s_1, T_1, is equal to the file length (in bytes) divided by the sum of the segment lengths, divided by the file streaming rate. It is easy to see from the equations in the Appendix that the file length and streaming rate affect each term in the required server bandwidth (C_{remote} or $C_{regional}$) only through the product of the file request rate, λ_i, and T_1. In each experiment below, total client request arrival rate is expressed in units of the average number of client requests that arrive per unit of time equal to the transmission time for segment s_1 (i.e., $R_1 = \lambda \times T_1$). Thus, the results hold for any file length and streaming rate that are in proportion to the specified constraints on proxy disk capacity and bandwidth. File selection frequencies, overall or within a group of files in the model defined in Section 3.2, are assumed to have the Zipf(0) distribution. In the graphs, the height of the black bar for each file (or object), in order of decreasing file popularity, indicates the percentage of the file that is stored at a given proxy server in a given minimum cost solution.

Table 4: Parameters Used in the Experiments

Parameter	Default Value	Parameter Definition
n	100	number of files
R_1	100	client request rate, measured in requests per unit of time equal to the transmission time for segment s_1
$N_{channels}$	288	maximum number of channels at each proxy server
$N_{segments}$	3654	storage capacity at each proxy server
P	10	number of proxy servers
k	7	number of segments in the leading segment set
K	12	total number of segments in the leading and trailing segment sets
l_j	1,2,2,4,4,8,8, 16,16,32,32,64	size of the j'th segment (relative to the size of the first segment)
W	64	the largest segment size (in the trailing segment set)

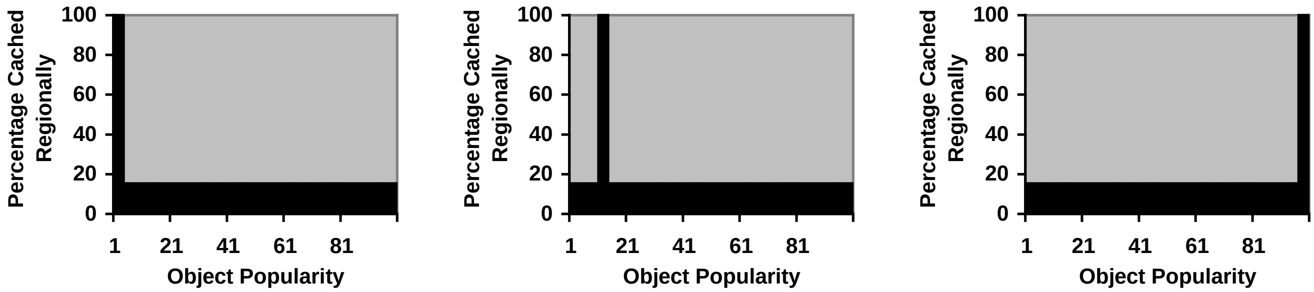
4.1 One Proxy Server with Light Request-Rate

The system considered here has ten regional servers and a total client request rate (R_1) equal to 100. One proxy server has (light) client request rate equal to 1.25; the other nine proxy servers each have identical (heavy) client request rate nearly equal to 11.

Figure 2 presents the socially optimal and individually optimal cache content at each type of proxy server when all servers have the same storage capacity and bandwidth, equal to the baseline values given in Table 4. The results show that the socially optimal cache content is the same for both the light and heavy proxy servers, and this is also the content that is individually optimal for each region. 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. Furthermore, the optimal cache content for each value of β is nearly identical to the optimal cache content for a homogeneous system with ten proxy servers that each have request rate equal to the heavy server request rate of ten (not shown).

Thus, when all proxy servers have the same bandwidth and storage capacity,

- the individually optimal content is the same as the socially optimal content for both light and heavy proxy servers,
- the optimal content for the heavy proxy servers is the same if the light server is not present, and
- the optimal content for the light proxy server is the same as the optimal content for the heavy servers.

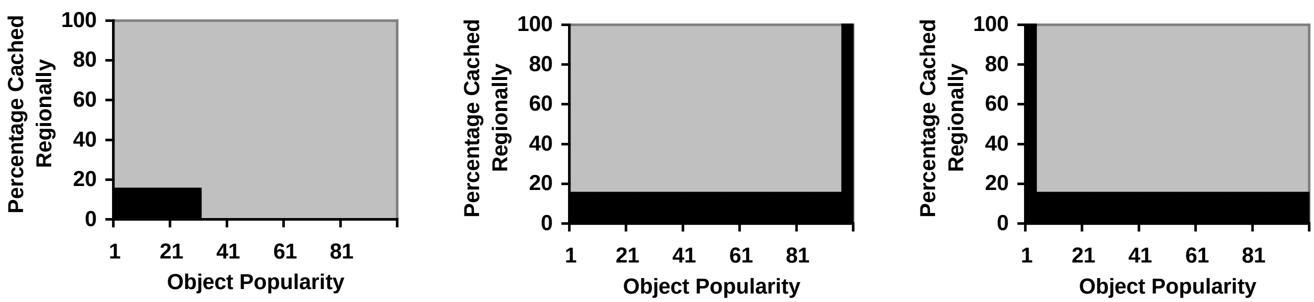


(a) β = 0, light & heavy social & individual

(b) β = 0.01, light & heavy social & individual

(c) β = 0.1, light & heavy social & individual

Figure 2: **Baseline System with One Light Server – Optimal Cache Content**
(P = 10, R_1 = 100, $R_{1,heavy} \approx 11$, $R_{1,light}$ = 1.25)



(a) β = 0 – 0.1, light proxy social & individual

(b) β = 0 – 0.1, heavy proxy socially optimal content

(c) β = 0, heavy proxy individual optimal

Figure 3: **One Light Server with Reduced Storage – Optimal Cache Content**
(P = 10, R_1 = 100, $R_{1,heavy} \approx 11$, $R_{1,light}$ = 1.25)

Figure 3 presents the optimal cache content when the proxy server with the lighter request rate has reduced storage and bandwidth, each equal to 25% of the baseline quantities. In this case:

- The socially optimal cache content differs for each type of proxy server, as shown in Figures 3a and 3b.
- For the proxy servers with greater storage and bandwidth (i.e., the servers with heavy request rate), the socially optimal cache content differs greatly from the individually optimal cache content, as shown in Figures 3b and 3c.[2]
- If the light request-rate server caches only the initial 15% of the most popular files, the socially optimal caching strategy for the heavy request rate proxy servers (shown in Figure 3b) is to *fully cache the least popular files, rather than the highly popular files* that they store in Figure 2a and 2b. This illustrates a very strong influence of the light server on the caching strategy at heavy servers, perhaps stronger than one might anticipate. The explanation for this impact is that the remote server must multicast the trailing segment set of the most popular files frequently for the light server. Thus, the greatest global advantage is for the heavy servers to share the cost of those multicasts and instead cache the less popular files that the light server needs very infrequently.

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 β equals 0.1, to minimize total delivery cost the light server caches exactly the same segments as the heavy servers; its extra storage and bandwidth capabilities are not used. However, the individually optimal solution for the light server, with the cached segments in the heavy servers fixed as they are for minimizing total delivery cost, shows that a competitive light server would store additional remote data. Thus, again, *the individually and socially optimal caching strategies differs dramatically for the server that has greater storage and bandwidth*.

4.2 One Proxy Server with Heavy Request-Rate

The first system considered here has eleven proxy servers, each with the baseline (i.e., default) bandwidth and storage capacity. Total client request rate is equal to 20. One proxy server's client load is equal to 10 requests per T_1; the other ten proxy servers have client request rate of 1.

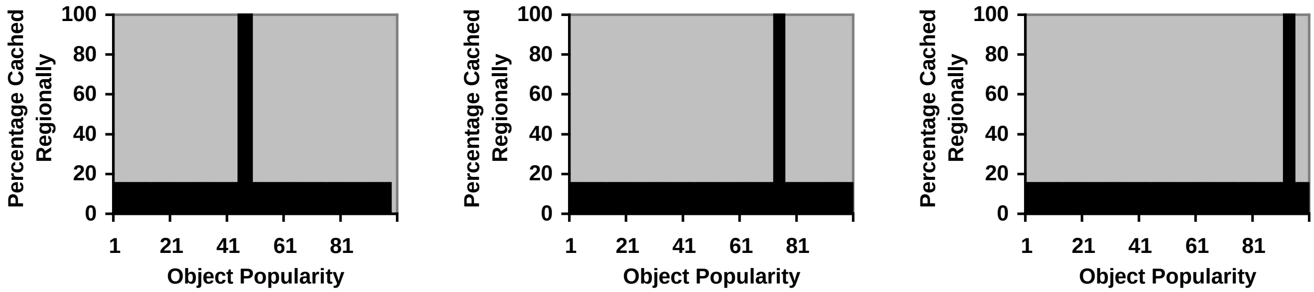
When β equals zero, the socially optimal and individually optimal cache content for all of the proxy servers is nearly identical to the content shown in Figure 2a. This content is in turn the optimal cache content in a homogeneous system with $\beta = 0$, any total client request rate greater than zero, and the same proxy storage capacity and bandwidth.

When β equal to 0.1, the socially optimal cache content, given in Figure 4b, is again the same for all of the proxy servers. However, the socially optimal cache content for the light servers is different than when the heavy server is not present (Figure 4a). Furthermore, it's interesting to note that files fully cached in the heterogeneous system (Figure 4b) are less popular than the files fully cached in Figure 4a, but are more popular than the files fully cached in the homogeneous system shown in Figure 4c. Figure 4c is the homogeneous system with total request rate equal to the total request rate in the heterogeneous system.

Minimizing the light server delivery cost while fixing segments cached at the heavy server as in Figure 4b, results in an individually optimal caching strategy that fully caches a slightly more popular set of files. 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 value of the optimization model is to discover such situations in which one might want to devise new pricing strategies.

We next consider a system in which the heavy request rate regional proxy server has *twice the baseline* server storage and bandwidth, while all other servers have the baseline storage and bandwidth. Overall client request rate is equal to 100. The heavy region's client request rate is equal to 20; each of the other nine regions' client request rate approximately equal to 9. The results are presented in Figure 5.

[2] If we minimize the individual delivery cost for the light server while fixing the cache content at the heavy servers as in Figure 3c, the light server caching strategy remains as in Figure 3a.

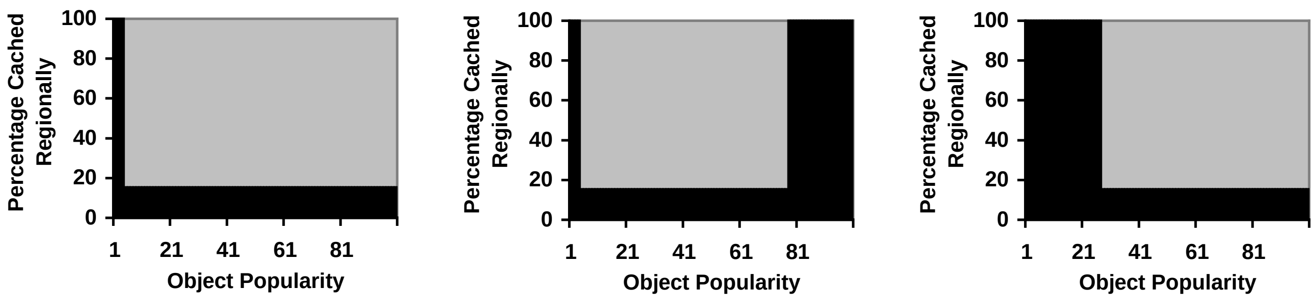


(a) β = 0.1, homogeneous
P = 10, R_1 = 10

(b) β = 0.1, heavy & light
socially optimal content

(c) β = 0.1, homogeneous
P = 10, R_1 = 20

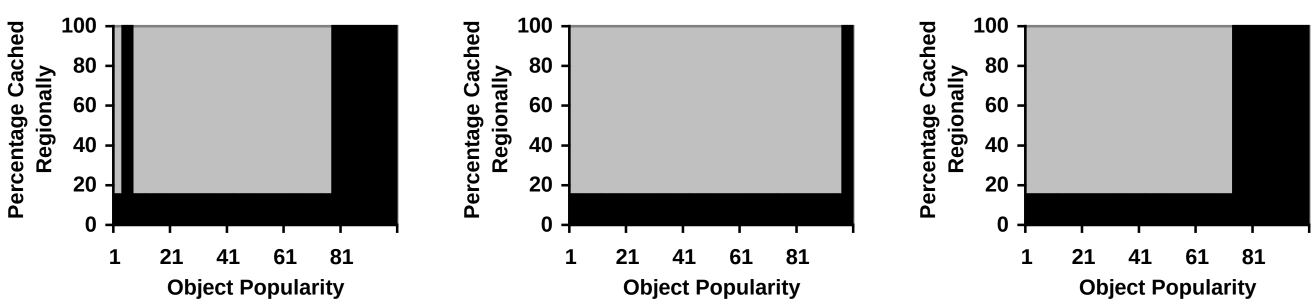
Figure 4: **Baseline System with One Heavy Server – Optimal Cache Content**
(P = 11 , R_1 = 20, $R_{1,\,heavy}$ = 10 , $R_{1,light}$ = 1)



(a) β = 0, light
social & individual

(b) β = 0, heavy
socially optimal

(c) β = 0, heavy
individually optimal



(d) β = 0.01, heavy
socially optimal

(e) β = 0.1, heavy & light
socially optimal

(f) β = 0.1, heavy
individualy optimal

Figure 5: One Heavy Server with Extra Storage – Optimal Cache Content
(P = 10, R_1 = 100, $R_{1,\,heavy}$ = 20 , $R_{1,light} \approx 9$)

For $0 \leq \beta \leq 0.1$, the socially optimal cache content for the proxy server with greater storage and bandwidth (e.g., Figures 5b and 5e) differs greatly from the individually optimal cache content for that server (e.g., Figures 5c and 5f, respectively). To minimize total delivery cost when $\beta = 0$, the heavy proxy server uses its extra storage to cache the least popular files (Figure 5b) rather than the most popular files (Figure 5c). To minimize total delivery cost when $\beta = 0.1$, the heavy proxy server stores exactly what the light proxy server stores (Figure 5e); it is more economical for the heavy rate server to share the remote streams that are delivered to the light regions than to use proxy server bandwidth to deliver the extra segments it could store (Figure 5f).

The results of this and the previous section indicate that proxy servers with light client request rates can greatly influence the socially optimal cache content for proxy servers with heavier client request rates if the storage and bandwidth at the light proxy servers is significantly less than the corresponding capabilities at the heavy servers.

4.3 Heterogeneous File Popularities

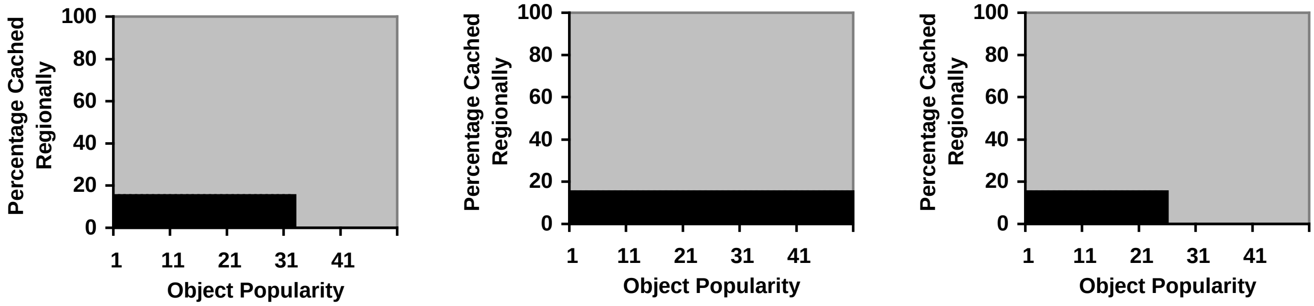
The next set of experiments considers systems with twelve regional servers partitioned into four groups of three servers, and two hundred files partitioned into four groups of fifty files. Each group of proxy servers has a different preferred group of files. The fraction of regional client requests to the preferred group is given by the parameter f_a; each non-preferred groups of files is selected with frequency $(1 - f_a)/3$. Selection frequency within each group (preferred or not) is modeled by a Zipf(0) distribution. The total client request rate, R_1, is equal to 100 requests per T_1.

Figure 6 provides results for the case that all proxy servers have the baseline storage capacity and bandwidth; Figure 7 provides results for the case that all proxy servers have three times the baseline storage capability. Figures 6a and 7a show the range of preferred group access frequencies under which all proxies store the same cache content. Figures 6b,c or 7b,c show the different optimal cache content for the preferred group and the other groups when the preferred group access frequencies are higher. Key conclusions from these results are:

- When preferred group access frequency is high enough, more segments are cached for the preferred group than for the other groups (Figure 6b compared with 6c, or Figure 7b compared with 7c).
- The threshold value of f_a above which the proxies store more data for their clients' preferred group, is higher when the proxy servers have greater bandwidth and storage capacity.
- When the proxies store more data for the preferred group, and the cache is large enough to fully cache some of the files (i.e., Figure 7b), the least popular preferred files are fully cached, rather than the most popular preferred files. This is again due to the advantages of sharing the cost of remote multicasts of the later segments of the most popular files.
- When proxy bandwidth and storage are limited (Figure 6), caching of initial segments of a subset of the files at all proxy servers appears to be a desirable compromise when different regions have different preferred files
- Similarity in proxy cache content among the heterogeneous servers is most critical for relatively more popular files.

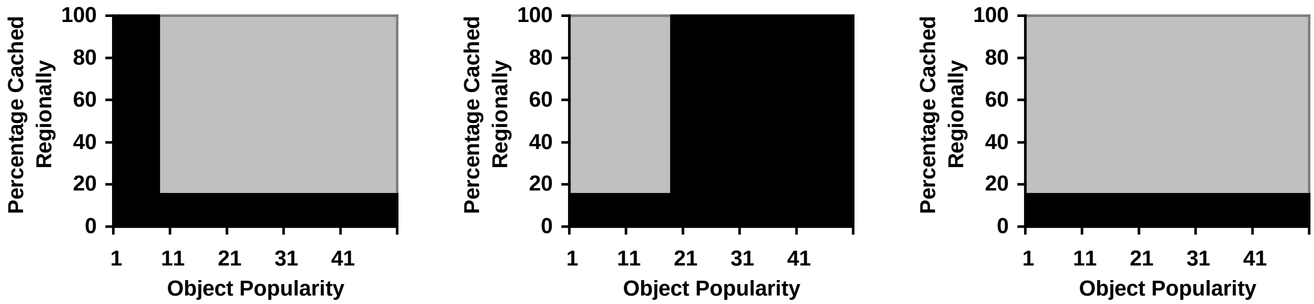
5 Conclusions

In this paper we have developed efficient analytic models for determining optimal caching strategies for continuous media data files in systems where the proxy servers have heterogeneous client populations and/or server capabilities. 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 remote file requests) across regional servers so as to achieve the greatest possible sharing of multicasts from a remote server. We have applied the models to obtain insights into how this conflict is resolved for specific types of heterogeneity. The results of those experiments indicate that:



(a) f_a = 0.25–0.4, all groups (b) f_a = 0.5–0.9, preferred group (c) f_a = 0.5–0.9, other groups

Figure 6: **Preferred File Group Per Region – Baseline System**
($\beta = 0$, $G = 4$, R_1 = 100)



(a) f_a = 0.25–0.75, all groups (b) f_a = 0.9, preferred group (c) f_a = 0.9, other groups

Figure 7: **Preferred File Group per Region – Three Times Baseline Proxy Server Capabilities**
($\beta = 0$, $G = 4$, R_1 = 100)

1. When the proxy cache cannot hold all of the popular files, it is almost always optimal to cache the initial segments of many files rather than all segments of fewer files, as was shown in Figures 2-6. In the case of heterogeneous file selection frequencies, there is an increased tendency to cache initial segments rather than full files, due to the improved cost-sharing for remote delivery when all servers cache the same segments.

2. Even in systems with quite strong heterogeneous features (e.g., Figures 2, 4b, 5e, 6a, and 7a), it is often more cost effective to store the same data at all of the proxy servers, rather than tailoring cache content at each proxy to the local client workload. The exceptions to this rule are when the proxy servers have different storage capacity and bandwidth (e.g., Figures 3a,c or Figures 5a,b) or when clients at different proxy servers have very strong preferences for different files (e.g., Figures 6b,c or Figures 7b,c).

3. When minimizing the total cost of delivery, a regional proxy server with a distinct client workload can sometimes influence the set of segments cached by other proxy servers in unexpected ways. In particular, a server with lower storage capacity or different preferred files can influence other servers to fully cache the least popular files rather than the most popular files (e.g., Figure 3b compared to 3c, Figure 5b compared to 5c, Figure 5e compared to 5f, or Figure 7b compared to 7c).

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. This occurred in particular for proxy servers that had greater storage capacity and bandwidth. Note that in such cases, it may be in the remote provider's best interest to encourage the regional proxy servers with less storage and bandwidth to increase their proxy server storage and bandwidth.

Due to the complex influences of heterogeneous client populations on the socially and individually optimal caching strategies at the proxy servers, it may be useful to consider deploying models in actual systems to compute optimal cache content for the measured client workloads.

On-going research includes: (1) extending the models in this paper to include more general heterogeneity in regional client workloads, (2) investigating optimal caching strategies for layered continuous media files and for multicast delivery techniques that permit more flexible partitioning of the data between the proxy and the remote server [13], and (3) devising on-line caching algorithms that achieve near-optimal cache content.

References

[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "A Permutation Based Pyramid Broadcasting Scheme for Video-on-Demand Systems", *Proc. IEEE Int'l. Conf. on Multimedia Computing and Systems (ICMCS'96)*, 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] Y. Cai, K. A. Hua, and K. Vu, "Optimizing Patching Performance", *Proc. IS&T/SPIE Conf. on Multimedia Computing and Networking 1999 (MMCN'99)*, San Jose, CA, January 1999, pp. 204-215.

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

[7] S. W. Carter and D. D. E. Long, "Improving Video-on-Demand Server Efficiency Through Stream Tapping", *Proc. 6th Int'l. Conf. on Computer Communications and Networks (ICCCN '97)*, Las Vegas, Nevada, September 1997, pp. 200-207.

[8] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrell, "A Hierarchical Internet Object Cache", *Proc. 1996 USENIX Technical Conf.*, January 1996.

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

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

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

[12] D. L. Eager, M. K. Vernon, and J. Zahorjan, "Minimizing Bandwidth Requirements for On-Demand Data Delivery", *Proc. 5th Int'l. Workshop on Multimedia Information Systems (MIS'99)*, Indian Wells, CA, October 1999, pp. 80-87.

[13] D. L. Eager, M. K. Vernon, and J. Zahorjan, "Optimal and Efficient Merging Schedules for Video-on-Demand Servers", *Proc. 7th ACM Int'l. Multimedia Conf. (ACM Multimedia'99)*, Orlando, FL, November 1999, pp. 199-202.

[14] D. L. Eager, M. K. Vernon, and J. Zahorjan, "Bandwidth Skimming: A Technique for Cost-Effective Video-on-Demand", *Proc. IS&T/SPIE Conf. on Multimedia Computing and Networking 2000 (MMCN 2000)*, San Jose, CA, January 2000.

[15] M. C. Ferris and R. R. Meyer, "Models and Solution for On-Demand Data Delivery Problems", in P. M. Pardalos (ed.), *Nonconvex Optimization and its Applications*, Vol. 42, Kluwer, 2000 (ftp://ftp.cs.wisc.edu/math-prog/tech-reports/99-04.ps).

[16] L. Gao, J. Kurose, and D. Towsley, "Efficient Schemes for Broadcasting Popular Videos", *Proc. 8th Int'l. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'98)*, Cambridge, UK, July 1998.

[17] L. Gao and D. Towsley, "Supplying Instantaneous Video-on-Demand Services Using Controlled Multicast", *Proc. 1999 IEEE Int'l. Conf. on Multimedia Computing and Systems (ICMCS'99)*, Florence, Italy, June 1999.

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

[19] K. A. Hua, Y. Cai, and S. Sheu, "Patching: A Multicast Technique for True Video-on-Demand Services", *Proc. 6th ACM Int'l. Multimedia Conf. (ACM Multimedia'98)*, Bristol, U.K., September 1998, pp 191-200.

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

[21] 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.

[22] J.-F. Paris, D. D. E. Long, and P. E. Mantey, "Zero-Delay Broadcasting Protocols for Video-on-Demand", *Proc. 7th ACM Int'l. Multimedia Conf. (ACM Multimedia'99)*, Orlando, FL, November 1999, pp. 189-197.

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

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

[25] S. Viswanathan and T. Imielinski, "Pyramid Broadcasting for Video-on-Demand Service", *Proc. IS&T/SPIE Conf. on Multimedia Computing and Networking 1995 (MMCN'95)*, San Jose, CA, February 1995, pp. 66-77.
[26] S. Viswanathan and T. Imielinski, "Metropolitan Area Video-on-Demand Service using Pyramid Broadcasting", *Multimedia Systems 4*, 4 (August 1996), pp. 197-208.

Appendix

Here we provide equations for computing the required remote server and proxy server bandwidths (C_{remote} and $C_{regional}$), and proxy server storage capacities ($D_{regional}$), as functions of the file request rates and the segments cached at the heterogeneous proxy servers.

A. Heterogeneous Client Request Rates and Server Capabilities

In the first heterogeneous system model described in the body of the paper, proxy servers are assumed to be identical except for a single proxy server that has a "distinct" (higher or lower) client request rate, and possibly also different server bandwidth and storage capacity. In addition to the notation defined in Tables 1 and 2, the detailed equations for this model also use quantities $X_{i,l,x}^{y,z}$ and $X_{i,t,x}^{y,z}$ $(x \in \{R, d, nd\}$, $y, z \in \{R, p, r\})$, which are defined as the maximum rates at which transmission clusters can be allocated for multicasts of the leading/trailing (*l/t*) segment set for file *i*, distinguished by the server (*R* – remote, *d* – distinct, or *nd* - other proxy), and by whether requests from a distinct (*y*) or non-distinct (*z*) region receive all of the file from the remote site (*R*), part (only the trailing segment set) from the remote site (*p*), or receive all of the file from the regional site (*r*). Further, as in the body of the paper, *w* denotes the largest segment size in the leading segment set, and T_1 denotes the duration of a transmission of segment s_1.

Since the leading and trailing segment sets are of duration wT_1 and WT_1, and include *k* and *K-k* channels, respectively, the average number of channels required at the remote server and at the two types of proxy servers, and the storage required at the proxy servers, are given by the following equations:

$$C_{remote}(\theta) = \sum_{i=1}^{n}\left(\sum_{y,z\in\{R,p\}} \theta_i^{y,z} X_{i,t,R}^{y,z} + \sum_{y\in\{R,p\}} \theta_i^{y,r} X_{i,t,R}^{y,r} + \sum_{z\in\{R,p\}} \theta_i^{r,z} X_{i,t,R}^{r,z} \right)(K-k)WT_1$$

$$+ \sum_{i=1}^{n}\left(\theta_i^{R,R} X_{i,l,R}^{R,R} + \sum_{y\in\{p,r\}} \theta_i^{y,R} X_{i,l,R}^{y,R} + \sum_{z\in\{p,r\}} \theta_i^{R,z} X_{i,l,R}^{R,z} \right) kwT_1$$

$$C_{regional}^{d}(\theta) = \sum_{i=1}^{n}\left(\sum_{z\in\{R,p,r\}} \theta_i^{r,z} X_{i,t,d}^{r,z}(K-k)WT_1 + \sum_{y\in\{p,r\}} \sum_{z\in\{R,p,r\}} \theta_i^{y,z} X_{i,l,d}^{y,z} kwT_1 \right)$$

$$C_{regional}^{nd}(\theta) = \sum_{i=1}^{n}\left(\sum_{y\in\{R,p,r\}} \theta_i^{y,r} X_{i,t,nd}^{y,r}(K-k)WT_1 + \sum_{y\in\{R,p,r\}} \sum_{z\in\{p,r\}} \theta_i^{y,z} X_{i,l,nd}^{y,z} kwT_1 \right)$$

$$D_{regional}^{d}(\theta) = \sum_{i=1}^{n} \sum_{z\in\{R,p,r\}} \left(\left(\theta_i^{r,z} + \theta_i^{p,z}\right) \sum_{j=1}^{k} l_j + \theta_i^{r,z} \sum_{j=k+1}^{K} l_j \right) \qquad D_{regional}^{nd}(\theta) = \sum_{i=1}^{n} \sum_{y\in\{R,p,r\}} \left(\left(\theta_i^{y,r} + \theta_i^{y,p}\right) \sum_{j=1}^{k} l_j + \theta_i^{y,r} \sum_{j=k+1}^{K} l_j \right)$$

The remote server channel cost that is apportioned to (clients of) the distinct proxy server is the sum of (a) 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:

$$C^{d}_{remote}(\theta)=\sum_{i=1}^{n}\left(\frac{\sum_{y,z\in\{R,p\}}\theta_i^{y,z}X_{i,t,R}^{y,z}}{P}+\sum_{y\in\{R,p\}}\theta_i^{y,r}X_{i,t,R}^{y,r}\right)(K-k)WT_1+\sum_{i=1}^{n}\left(\frac{\theta_i^{R,R}X_{i,l,R}^{R,R}}{P}+\sum_{z\in\{p,r\}}\theta_i^{R,z}X_{i,l,R}^{R,z}\right)kwT_1$$

Similarly, the remote server channel cost apportioned to (clients of) a non-distinct proxy server is given by:

$$C^{nd}_{remote}(\theta)=\sum_{i=1}^{n}\left(\frac{\sum_{y,z\in\{R,p\}}\theta_i^{y,z}X_{i,t,R}^{y,z}}{P}+\frac{\sum_{z\in\{R,p\}}\theta_i^{r,z}X_{i,t,R}^{r,z}}{P-1}\right)(K-k)WT_1+\sum_{i=1}^{n}\left(\frac{\theta_i^{R,R}X_{i,l,R}^{R,R}}{P}+\frac{\sum_{y\in\{p,r\}}\theta_i^{y,R}X_{i,l,R}^{y,R}}{P-1}\right)kwT_1$$

The maximum rates X_i at which transmission clusters can be allocated are determined by the arrival rate of requests at the server of interest, and by the size of the transmission cluster "catchup window" (defined as the period of time from the beginning of a transmission cluster during which a newly arriving request can be served by that cluster). Transmission clusters for leading and trailing segment sets have catchup windows of differing lengths. Further, the length of the catchup window for a leading segment set transmission cluster depends on when the cluster begins in relationship to the end of the catchup window of the corresponding trailing segment set transmission cluster. For each distinct case, the rate X_i may be computed as the inverse of the minimum average time between the initiations of new clusters for file *i*.

For example, consider the case in which all clients receive the trailing segment set of file *i* from the remote site. Since the size of the catchup window for a trailing segment set transmission cluster is $\left(W-l_{k+1}+\sum_{j=1}^{k}l_j\right)T_1$, and since the average time from the end of the catchup window until a new client request for file *i* is $1/\lambda_i$, the maximum allocation rate of new transmission clusters for the trailing segment set of file *i* is computed as:

$$X_{i,t,R}^{R|p,R|p}=\frac{1}{\frac{1}{\lambda_i}+\left(W-l_{k+1}+\sum_{j=1}^{k}l_j\right)T_1}$$

In this equation, and those given below, the superscript notations "*" ("wild-card"), "-" ("don't care"), and "|" ("or") are employed with the following meanings: an equation in which a "*" superscript appears holds for any consistent substitution throughout the equation (in that superscript position) of the "*" with *R*, *p*, or *r*; an equation in which a "-" superscript appears holds for any (not necessarily consistent) substitution (in that superscript position) of the "-" with *R*, *p*, or *r*; and an equation in which a superscript "*x*|*y*" appears, where $x,y\in\{R,p,r\}$, holds if either *x* or *y* is consistently substituted throughout the equation (in that superscript position).

Note that the above expression gives the allocation rate of transmission clusters if the server had unlimited bandwidth. Since the server has limited bandwidth, the above expression is the maximum allocation rate, as there may be queueing of client requests and batching of these requests while queued, or client balking.

The remaining X_i equations for the model with heterogeneous client request rates and server capabilities are as follows:

$$X_{i,t,R}^{R|p,r}=\frac{1}{\frac{1}{f_d\lambda_i}+\left(W-l_{k+1}+\sum_{j=1}^{k}l_j\right)T_1}\qquad X_{i,t,R}^{r,R|p}=\frac{1}{\frac{1}{(1-f_d)\lambda_i}+\left(W-l_{k+1}+\sum_{j=1}^{k}l_j\right)T_1}$$

$$X_{i,t,d}^{r,-}=\frac{1}{\frac{1}{f_d\lambda_i}+\left(W-l_{k+1}+\sum_{j=1}^{k}l_j\right)T_1}\qquad X_{i,t,nd}^{-,r}=\frac{1}{\frac{P-1}{(1-f_d)\lambda_i}+\left(W-l_{k+1}+\sum_{j=1}^{k}l_j\right)T_1}$$

$$X_{i,l,R}^{p|r,R} = \frac{1}{\frac{1}{(1-f_d)\lambda_i} + \left(X_{i,t,R}^{p|r,R} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{p|r,R} wT_1)(w-1) \right) T_1} \qquad X_{i,l,R}^{R,R} = \frac{1}{\frac{1}{\lambda_i} + \left(X_{i,t,R}^{R,R} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{R,R} wT_1)(w-1) \right) T_1}$$

$$X_{i,l,R}^{R,p|r} = \frac{1}{\frac{1}{f_d \lambda_i} + \left(X_{i,t,R}^{R,p|r} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{R,p|r} wT_1)(w-1) \right) T_1} \qquad X_{i,l,d}^{r,-} = \frac{1}{\frac{1}{f_d \lambda_i} + \left(X_{i,t,d}^{r,-} wT_1 \frac{w-1}{2} + (1 - X_{i,t,d}^{r,-} wT_1)(w-1) \right) T_1}$$

$$X_{i,l,d}^{p,*} = \frac{1}{\frac{1}{f_d \lambda_i} + \left(X_{i,t,R}^{p,*} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{p,*} wT_1)(w-1) \right) T_1} \qquad X_{i,l,nd}^{-,r} = \frac{1}{\frac{P-1}{(1-f_d)\lambda_i} + \left(X_{i,t,nd}^{-,r} wT_1 \frac{w-1}{2} + (1 - X_{i,t,nd}^{-,r} wT_1)(w-1) \right) T_1}$$

$$X_{i,l,nd}^{*,p} = \frac{1}{\frac{P-1}{(1-f_d)\lambda_i} + \left(X_{i,t,R}^{*,p} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{*,p} wT_1)(w-1) \right) T_1}$$

B. Heterogeneous File Selection Frequencies

In the second heterogeneous system model described in the body of the paper, the files 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 files. Each proxy server has the same request rate for each of its (G-1) non-preferred groups of files. Also, the *relative* selection frequencies of the files *within a group* are the same for all groups and for all proxy servers. In addition to the notation defined in Tables 1 and 3, the detailed equations for this model also use quantities $X_{i,l,x}^{y,z}$ and $X_{i,t,x}^{y,z}$ ($x \in \{R, a, na\}$, $y, z \in \{R, p, r\}$), which are defined as the maximum rates at which transmission clusters can be allocated for multicasts of the leading/trailing (l/t) segment set for file i, distinguished by the server (R – remote, a – a proxy server that has a preference or *affinity* for the file's group, or na – a proxy server that does not have a preference for the file's group), and by whether requests from a preferring (y) or non-preferring (z) region receive all of the file from the remote site (R), part (only the trailing segment set) from the remote site (p), or receive all of the file from the regional site (r). Note that although each proxy server may optimally cache different file 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.

The number of channels required at the remote server and at each proxy server, and the storage required at each proxy server, are given by the equations below, where the index i runs only over the files within a single group since the groups of files are symmetric:

$$C_{remote}(\theta) = G \sum_i \left(\sum_{y,z \in \{R,p\}} \theta_i^{y,z} X_{i,t,R}^{y,z} + \sum_{y \in \{R,p\}} \theta_i^{y,r} X_{i,t,R}^{y,r} + \sum_{z \in \{R,p\}} \theta_i^{r,z} X_{i,t,R}^{r,z} \right) (K-k) WT_1$$

$$+ G \sum_i \left(\theta_i^{R,R} X_{i,l,R}^{R,R} + \sum_{y \in \{p,r\}} \theta_i^{y,R} X_{i,l,R}^{y,R} + \sum_{z \in \{p,r\}} \theta_i^{R,z} X_{i,l,R}^{R,z} \right) kwT_1$$

$$C_{regional}(\theta) = \sum_i \left(\sum_{z \in \{R,p,r\}} \theta_i^{r,z} X_{i,t,a}^{r,z} + (G-1) \sum_{y \in \{R,p,r\}} \theta_i^{y,r} X_{i,t,na}^{y,r} \right) (K-k) WT_1$$

$$+\sum_i \left(\sum_{y\in\{p,r\}} \sum_{z\in\{R,p,r\}} \theta_i^{y,z} X_{i,l,a}^{y,z} + (G-1) \sum_{y\in\{R,p,r\}} \sum_{z\in\{p,r\}} \theta_i^{y,z} X_{i,l,na}^{y,z} \right) kwT_1$$

$$D_{regional}(\theta) = \sum_i \sum_{z\in\{R,p,r\}} \left(\left(\theta_i^{r,z} + \theta_i^{p,z}\right) \sum_{j=1}^{k} l_j + \theta_i^{r,z} \sum_{j=k+1}^{K} l_j \right) + (G-1) \sum_i \sum_{y\in\{R,p,r\}} \left(\left(\theta_i^{y,r} + \theta_i^{y,p}\right) \sum_{j=1}^{k} l_j + \theta_i^{y,r} \sum_{j=k+1}^{K} l_j \right)$$

The maximum rates X_i at which transmission clusters can be allocated are computed in a similar fashion as in the previous models. Using the "wildcard", "don't care", and "or" notation used previously for the request rate and server heterogeneity model, we have:

$$X_{i,t,R}^{R|p,R|p} = \frac{1}{\frac{1}{\lambda_i} + \left(W - l_{k+1} + \sum_{j=1}^{k} l_j \right) T_1} \qquad X_{i,t,R}^{R|p,r} = \frac{1}{\frac{1}{f_a \lambda_i} + \left(W - l_{k+1} + \sum_{j=1}^{k} l_j \right) T_1}$$

$$X_{i,t,R}^{r,R|p} = \frac{1}{\frac{1}{(1-f_a)\lambda_i} + \left(W - l_{k+1} + \sum_{j=1}^{k} l_j \right) T_1} \qquad X_{i,t,a}^{r,-} = \frac{1}{\frac{P}{G f_a \lambda_i} + \left(W - l_{k+1} + \sum_{j=1}^{k} l_j \right) T_1}$$

$$X_{i,t,na}^{-,r} = \frac{1}{\frac{G-1}{G} \frac{P}{(1-f_a)\lambda_i} + \left(W - l_{k+1} + \sum_{j=1}^{k} l_j \right) T_1} \qquad X_{i,l,R}^{p|r,R} = \frac{1}{\frac{1}{(1-f_a)\lambda_i} + \left(X_{i,t,R}^{p|r,R} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{p|r,R} wT_1)(w-1) \right) T_1}$$

$$X_{i,l,R}^{R,R} = \frac{1}{\frac{1}{\lambda_i} + \left(X_{i,t,R}^{R,R} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{R,R} wT_1)(w-1) \right) T_1} \qquad X_{i,l,R}^{R,p|r} = \frac{1}{\frac{1}{f_a \lambda_i} + \left(X_{i,t,R}^{R,p|r} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{R,p|r} wT_1)(w-1) \right) T_1}$$

$$X_{i,l,a}^{r,-} = \frac{1}{\frac{P}{G f_a \lambda_i} + \left(X_{i,t,a}^{r,-} wT_1 \frac{w-1}{2} + (1 - X_{i,t,a}^{r,-} wT_1)(w-1) \right) T_1} \qquad X_{i,l,a}^{p,*} = \frac{1}{\frac{P}{G f_a \lambda_i} + \left(X_{i,t,R}^{p,*} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{p,*} wT_1)(w-1) \right) T_1}$$

$$X_{i,l,na}^{-,r} = \frac{1}{\frac{G-1}{G} \frac{P}{(1-f_a)\lambda_i} + \left(X_{i,t,na}^{-,r} wT_1 \frac{w-1}{2} + (1 - X_{i,t,na}^{-,r} wT_1)(w-1) \right) T_1}$$

$$X_{i,l,na}^{*,p} = \frac{1}{\frac{G-1}{G} \frac{P}{(1-f_a)\lambda_i} + \left(X_{i,t,R}^{*,p} wT_1 \frac{w-1}{2} + (1 - X_{i,t,R}^{*,p} wT_1)(w-1) \right) T_1}$$