Cooperative Cashing?: An Economic Analysis of Document Duplication in Cooperative Web Caching

Abstract

Cooperative caching is a popular mechanism to allow an array of distributed caches to cooperate and serve each others' web requests. Controlling duplication of documents across cooperating caches is a challenging problem faced by cache managers. In this paper, we study the economics of document duplication in strategic and non-strategic settings. We have three primary findings. First, we find that the optimum level of duplication at a cache is non-decreasing in inter-cache latency, cache size and extent of request locality. Second, in situations in which cache peering spans organizations, we find that the interaction between caches is a game of strategic substitutes wherein a cache employs lesser resources towards eliminating duplicate documents when the other caches employs more resources towards eliminating duplicate documents at that cache. Thus, a significant challenge will be to simultaneously induce multiple caches to contribute more resources towards reducing duplicate documents in the system. Finally centralized decision-making, which as expected provides improvements in average latency over a decentralized setup, can entail highly asymmetric duplication levels at the caches. This in turn can benefit one set of users at the expense of the other and thus will be challenging to implement.

Keywords: web caching, cooperative caching, duplication in caching, analytical modeling, incentive centered design, game theory.

1. Introduction

Web caching refers to the temporary storage of web content somewhere between web servers and clients in order to satisfy future requests from the nearby location (see Figure 1). Proxy caches, located at the gateways of large organizations and ISPs, play an important role in reducing latency (i.e., delay in content delivery) and bandwidth costs. Forrester Research prescribes caching as one of four best practices to improve performance for websites and ISPs (Gualteiri and Staten 2009).

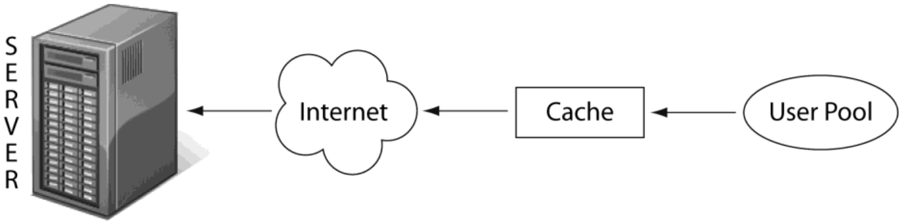

Figure 1: A Proxy Web Cache

Often, multiple caches in a network coordinate and share resources in order to serve each others' requests (see Figure 2). This is also known as cooperative caching. When a cache does not have the requested data object, it can forward the request to a nearby cooperating cache that can serve the object faster than the origin server. The primary benefits of cache cooperation are higher hit rates[1] and lower average latency for end users. Cooperative caching is typically implemented across caches within an organization such as a large enterprise, ISP, or a Content Delivery Network (CDN). Cache cooperation can sometimes span organizational boundaries, for example with cache peering at exchanges such as Packet Clearing House and Equinix as well as implementations in the public domain such as IRCache and w3cache.[2]

[1] Hit rate or hit ratio indicates the fraction of requests served from the cache.
[2] www.pch.net, www.equinix.com, www.ircache.net, http://w3cache.icm.edu.pl.

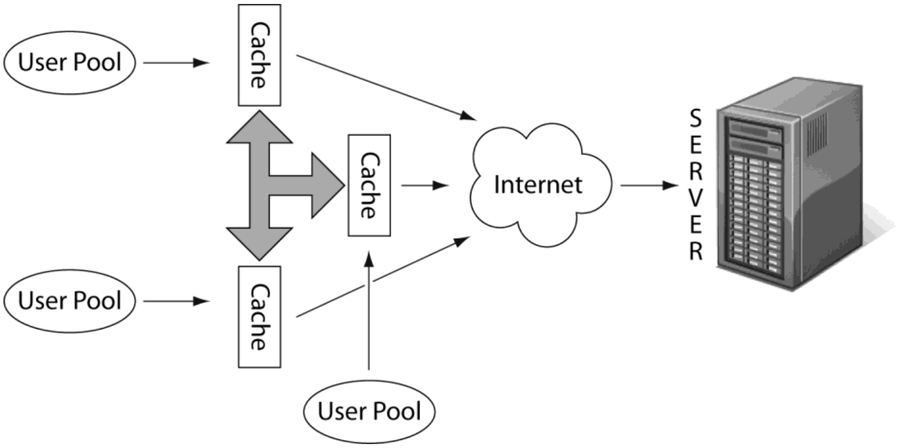

Figure 2: Cooperative Caching

A number of protocols have been proposed to help determine how caches should communicate and coordinate. These include Inter Cache Protocol (ICP), Cache Array Routing Protocol (CARP), Summary Cache and Home Protocol. ICP (Wessels and Claffy 1998), one of the first cooperative caching protocols to be supported in commercial proxy servers, lets each cache independently determine what objects to cache without accounting for content in other caches. This eliminates the need to coordinate but can result in low array hit ratio because of duplication of objects across the caches. CARP (Valloppillil and Ross 1998), a protocol first used in Microsoft Proxy Server, assigns every data object to a designated cache based on a hash function. Each cache stores only those objects that it is designated to cache. This maximizes the number of objects collectively stored by the cache array. However, it can result in a low local hit ratio, e.g., when a popular object is always fetched from a remote cache. Ultimately, the end user is concerned with latency, which is a function of both local and array hit rates. To achieve a better balance, protocols that set duplication levels in between that achieved under ICP and CARP have also been proposed. Using trace-driven simulations, studies show that these protocols help balance the load of the caches and reduce user latency by 10-20% (Wu and Yu 1999a, Zu and Subhlok 2003). Given the non-trivial gains realizable from tuning the duplication level, this is a key factor influencing the returns from cache cooperation.

Recognizing that neither ICP nor CARP will consistently outperform the other, most proxy servers including Squid and Sun Java System proxy server currently support both ICP and CARP and let the system administrator select the protocol at deployment. System administrators can also set intermediate duplication levels using recent techniques which, although not currently supported in commercial proxy servers, can be implemented with some additional effort such as through the logical partitioning of caches (Wu and Yu 1999a). The absence of a default setting in most proxy servers is because the optimal choice depends on the deployment context. This provides significant flexibility to administrators but also imposes the additional burden of selecting the right level of duplication. Considerable experimentation is often needed to determine the appropriate choice.[3] The trade and academic press provide prescriptions ranging from "CARP versus ICP: ... CARP is a genuine evolution of ICP, providing much better scalability and performance" (Northrup 1998) to "ICP (has) considerably lower response time than CARP. The reason is the low local hit ratio of CARP." (Zu and Subhlok 2003). Simultaneously, recent academic studies suggest intermediate levels of duplication are desirable. These studies are based on simulations of distributed non-strategic caches and the diversity of their conclusions reflects the diversity of the simulation settings. They do not develop a theory to help understand the fundamental tradeoffs in determining duplication levels. Further they raise additional questions for the policy-maker. For example, when should the cache manager use configurations that result in no duplication, unmonitored duplication or some intermediate level of duplication? In case of cache peering across organizations, what is the impact of strategic behavior on equilibrium duplication levels and latency? We conduct an economic analysis of document duplication to address these questions.

We study the problem in two ways. First, we develop analytical models to study document duplication in cooperative caching.[4] We make several assumptions that help specify a tractable model from which insights regarding the impact of various parameters can be derived. Second, we turn to trace-driven simulations to validate our findings under more realistic settings. The novelty of our analytic

[3] Industry discussion forums provide some examples: "I have gone back and forth on what protocol to use in order to maximize the efficiency of my squid cache cluster (HTCP, ICP, CARP)". Retrieved August 2008 from http://www.nabble.com/Carp-is-resulting-in-403s-td18843567.html. See also http://www.nabble.com/cache-hierarchy-question-td18919695.html.

[4] Our study focuses on document duplication rather than the broader question of protocol selection. Protocol selection involves multiple considerations beyond duplication levels. We discuss some of these considerations in Section 2.

approach lies in the fact that we model the request process and caching decisions for the full set of documents in a cache to ultimately capture the impact of document duplication on expected latency. As a result, the economic model captures the operational details of cooperative caching protocols.

Our main contribution is the development of a formal framework with which to analyze the tradeoffs associated with document duplication in cooperative web caching. We arrive at three primary findings. First, we find that the optimum level of duplication is non-decreasing in inter-cache latency, cache size and extent of request locality.[5] Correspondingly, zero duplication is preferable to unmonitored duplication when caches are close by and are smaller in size and requests exhibit low locality, and vice-versa. Second, in situations in which cache peering spans organizations, we find that the interaction between caches is a game of strategic substitutes wherein a cache employs lesser resources towards eliminating duplicate documents when the other caches employ more resources towards eliminating duplicate documents. Thus, a significant challenge will be to simultaneously induce multiple caches to allocate resources towards reducing document duplication in the cache array. Finally centralized decision-making, as expected, provides improvements in latency over a decentralized setup. But more significantly, it can entail highly asymmetric duplication levels at the caches. This in turn penalizes some users while benefiting others and thus raises implementation challenges.

The rest of the paper is organized as follows. In Section 2, we review the related literature. In Section 3, we develop a model to study optimal duplication in a setting with two caches. We apply the model to a variety of decision contexts, including decentralized and centralized decision contexts. In Section 4, we test the robustness of our results in two parts. First, we extend our analytical model and relax two key assumptions. Next, we use trace-driven simulations to validate our theoretical findings under a realistic environment. We conclude the study in Section 5.

2. Literature Review

[5] Locality in web caching refers to the fact that a recently requested document is likely to be requested again.

Web caching has been a popular research stream in Computer Science and Information Systems. The two streams of work most relevant to our paper are the ones on a) cooperative caching protocols with a particular emphasis on document duplication, and b) Management Science research on caching.

Document Duplication in cooperative Caching: A number of protocols have been proposed to address cache coordination in cooperative caching. These include ICP, CARP, Summary Cache, Home and Two-exit LRU among others.

In ICP (Wessels and Claffy 1998), each cache independently determines which objects to cache (for e.g., each cache uses LRU).[6] Whenever there is a local miss, queries are sent to all other caches in the array. If any of the caches have the content, they respond with the content else the request is forwarded to the origin server. A major disadvantage of ICP is that a large number of queries are sent, especially if the number of caches in the array is large. Summary Cache (Fan et al 1998) and Cache Digest (Rousskov and Wessels 1998) address this by maintaining an index/directory of current content in all the caches. A local miss results in a lookup of the index, and the request is forwarded to the relevant cache. Several studies show that the indexes can be maintained with relatively low overheads if caches delay propagation of directory updates (Fan et al. 1998, Tewari et al. 1999). There is a high degree of duplication of objects in ICP, Summary Cache and Cache Digest because the caches do not coordinate which objects to store. This results in low array hit ratio.

In CARP (Valloppillil and Ross 1998), every data object is assigned to a designated cache based on a hash function. Each cache stores only the objects it is designated to cache. This helps ensure that the maximum number of data objects is collectively stored in the array resulting in a high array hit ratio. Furthermore, a local miss results in the request being forwarded to the designated cache alone which implies lower query traffic. However, a primary disadvantage of CARP is that it has a low local hit ratio. For example, a popular document may be assigned to another cache resulting in a local miss whenever the document is requested locally.

[6] Least Recently Used (LRU) is a replacement policy commonly used in web caches. A replacement policy specifies the object that should be removed from a cache whenever a new object is added to a cache. LRU replaces the object that was least recently requested assuming that it is least likely to be requested again among the objects in the cache.

Given the deficiencies of unmonitored duplication and zero duplication, recent research has proposed allowing some amount of duplication of documents across caches. Two-exit LRU (Wu and Yu 1999a), Adaptable Controllable Replication (Wu and Yu 1999b), Home Protocol (Zu and Subhlok 2003) and Hosanagar and Tan (2004) all allow a cache to store its most popular documents irrespective of its presence in one or more other caches. These schemes often divide a cache into two regions as shown in Figure 3. In the duplication region, a regular LRU scheme is used to store the most popular documents independent of whether these documents also exist in the other caches. In the non-duplication region, the cache will not store a document if it exists in another cache. The decision of whether to store a document in the non-duplication region can be made using a hash function (Wu and Yu 1999a, 1999b) or by using an index to confirm the document is not in another cache (Hosanagar and Tan 2004).

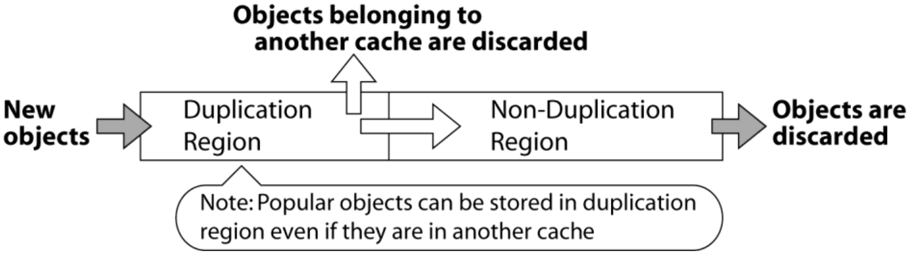
Objects belonging to another cache are discarded
New objects
Duplication Region
Non-Duplication Region
Objects are discarded
Note: Popular objects can be stored in duplication region even if they are in another cache

Figure 3: Controlling Duplication by Logical Partitioning of Caches

In summary, the protocols differ along two main dimensions – communication overhead and level of duplication (see Table 1). In terms of communication overhead, ICP broadcasts queries to all caches in the array whenever there is a local miss whereas the other protocols query caches in a targeted manner through either the use of directories or URL hashing. ICP is rarely used when there are a large number of caches in the array because of the communication overhead of broadcasting to all caches. When the array size is relatively manageable, this overhead of ICP is less of an issue. In terms of duplication, ICP, Summary Cache and Cache Digest do not monitor duplication levels, CARP does not permit duplication, and Two-exit LRU and Home can set duplication to any value in between.

Our paper focuses exclusively on the level of duplication and we do not evaluate overhead costs. This is partly because it is possible to use URL hashing or directories to achieve low communication overhead regardless of the level of duplication. For example, Two-exit LRU uses URL hashing to reduce communication overhead but can implement zero duplication and unmonitored duplication as special cases. Further, unlike communication overhead, of which less is always better, there is no universally preferred level of duplication. Therefore, we focus on the optimal level of duplication and investigate how it depends on various parameters. These arguments notwithstanding, a natural extension of our work is to focus on the issue of protocol selection by considering duplication levels, communication and storage overheads and other factors.

	ICP	Summary cache	Cache Digest	CARP	Two-exit LRU	Home
Communication with other caches	Broadcast	Targeted (Directory-based)	Targeted (Directory-based)	Targeted (URL Hashing)	Targeted (URL Hashing)	Targeted (URL Hashing)
Duplication	Unmonitored	Unmonitored	Unmonitored	Zero	Intermediate	Intermediate

Table 1: Comparison of cooperative caching protocols

It is also worth noting that although many of the papers described above use simulations to demonstrate that controlled duplication often helps improve latency, they neither prescribe the optimal level of duplication nor do they provide a framework to understand the factors driving the optimal duplication levels. Further, even though cooperative caching is a decentralized process, they do not consider the strategic behavior of the caches. Our model complements these papers by developing a theoretical framework to better understand the tradeoffs in determining document duplication and the impact of strategic behavior on equilibrium outcomes. These insights can help cache operators select the best approach when deploying a network of peering caches.

Management Science Research on Web Caching: There has been a lot of recent interest in web caching in the Information Systems (IS)/ Management Science (MS) community. For a detailed overview of web caching from a Management Science perspective, we refer the reader to Datta et al. (2003). One important stream of work has focused on modeling the key operational decisions at proxy caches and demonstrated that performance improvements can be achieved through careful optimization. The key operational decisions at proxy caches include the rules to determine which objects to add to a cache (placement policy) and rules to determine which objects to remove when a new object is added (replacement policy). Fang et al. (2006) propose and test a pre-fetching technique for caches in a network storage system to pre-fetch objects that users are likely to request in the near future. Dutta et al. (2006) and Chiang et al. (2007) formulate models for identifying objects/fragments to cache and the frequency with which they should be replaced. Mookerjee and Tan (2002) analyze the performance of an LRU policy for browser caches. Kaya et al. (2009) propose an admission-control policy for proxy server caching that augments the LRU mechanism. Kumar and Norris (2008) propose a mechanism that takes into account aggregate patterns in user object requests and show that it can outperform LRU. Bose and Cheng (2000) show that proxy caching is beneficial if the hit rate exceeds a threshold, and identify the factors on which the threshold depends.

The stream of work that is closely related to our paper is that studying the economics and operational aspects of distributed caching. Chan et al. (1999) and Chuang and Sirbu (2000) propose markets for QoS-based caching services and Hosanagar et al. (2005) study the design and pricing of these services. Geng et al (2003) also discuss a cooperative caching market in which ISPs may trade cache capacity. More recently, Du et al (2008) address the viability of a cache coordination network coordinated through an allocation hub. Cache networks can also be deployed by a central provider such as a Content Delivery Network (CDN). Dogan et al. (2003) and Hosanagar et al. (2008) study pricing of CDNs that maintain a network of cooperating caches. In terms of operational issues, Tan et al (2006) develop models for coordinating object placement decisions between browser and proxy-server caches, and Tawarmalani et al (2008) and Kumar (2009) formulate and solve nonlinear programs to allocate objects in a cache array. The technologies and market mechanisms that facilitate distributed caching within and across organizational boundaries have clearly been of much interest to the Management Science community. Our paper contributes to this stream of work by studying the economics of document duplication in cooperative caching.

3. Analytical Model

We begin by stating our assumptions and introducing our notation. We analyze the case of two caches cooperating with each other. The caches are heterogeneous in terms of their sizes. Requests to the caches are independent. All documents are of the same size and server download time for all the documents is the same. This assumption is for analytical tractability. In Section 4, we relax our assumptions and test the robustness of our findings using trace-driven simulations.

Request arrival at a cache is assumed to follow an inhomogeneous Poisson process. That is, the request arrivals for any document follows a Poisson distribution but the mean arrival rates vary based on the time since last request for the object. Although a Poisson arrival process at caches is commonly assumed in the literature (e.g., Che et al 2002), it ignores locality in web requests. We use an inhomogeneous Poisson process to address this shortcoming. In our model, the mean instantaneous access rate for a document with LRU age (time since the last request) *x* is assumed to be

$$\theta(x) = \frac{1}{\alpha \cdot x + \beta}; \quad \alpha, \beta > 0; \alpha < 1 \tag{1}$$

The parameter α measures the sensitivity of the access rate to the LRU age. A high value of α implies that LRU age is a good predictor of future requests i.e., there is high locality in requests. Setting $\alpha = 0$ models a regular Poisson process. With an inhomogeneous Poisson process, the probability density of a document with age *x*, i.e. the probability of having a document with LRU age *x*, (Tan et al 2006) is,

$$f(x) = \beta^{\frac{1-\alpha}{\alpha}}(1-\alpha)(\alpha \cdot x + \beta)^{-\frac{1}{\alpha}}, \quad x > 0 \tag{2}$$

The total mean access rate for the n documents (Tan et al 2006) is :

$$H_0 = n\int_0^\infty \theta f(x)dx = \frac{1-\alpha}{\beta}n \tag{3}$$

At any given instant, we can rank order the n documents based on their LRU age in a proxy cache. With a simple LRU policy, a cache of size *R* would have stored the top *R* documents. In the new scheme, the cache is divided into a duplication region of size *L* and a non-duplication region of size (*R* − *L*). The cache can store any document in the duplication region but no duplication is allowed in the non-duplication region. Thus, the *L* most popular documents are stored in the duplication region regardless of whether these documents are in the other cache. However, document (*L* + 1) is stored in the cache only if it is not already present in the other cache. Similarly for documents (L + 2) and onwards, the cache stores only those documents that are not present in the other cache. The advantage of this framework is that it incorporates zero duplication and unmonitored duplication as special cases (L=0 and L=R respectively). The setup is similar to the Two-exit LRU scheme proposed by Wu and Yu (1999a, 1999b) and a variant discussed by Hosanagar and Tan (2004). Wu and Yu (1999) use a hash function like in CARP to determine which objects to store in the non-duplication region. Hosanagar and Tan (2004) adopt a variant that does not use a hash function but instead verifies an object is not in another cache by consulting a directory/index as in Summary Cache. We sketch the implementation of both these schemes in the online appendix. To fix a context and also because it is harder to model hash functions analytically, our model below assumes a directory-based approach. Although we study the tradeoffs under the directory-based approach, we expect that the fundamental tradeoffs tied to duplication do not depend on the specific mechanism used to coordinate the non-duplication region.

Consider the configuration of cache 2 at some arbitrary instant. The indicator variable φ_k denotes whether document k exists in cache 1. That is, document k is tagged $\varphi_k = 1$ if it is in cache 1 and $\varphi_k = 0$ otherwise. If we rank order the documents by their LRU age in cache 2, the first L_2 documents are stored in cache 2 irrespective of the value of φ_k for these documents. Let us denote the number of tagged documents among the first L_2 documents by j. That is, $\sum_{k=1}^{L_2} \varphi_k = j$. For document ($L_2$ + 1) and higher, the document is stored only if $\varphi_k = 0$. Let i denote the number of documents that are skipped (i.e., not stored in cache 2 because they exist in cache 1) before the ($R_2 - L_2$)*th* document is encountered for the no-duplication region. That is, the final document that is stored in cache 2 is the (R_2 + i)*th* document. Thus, $\sum_{k=L_2+1}^{R_2+i} \varphi_k = i$. It is clear that if the size of the duplication region is reduced, the two caches collectively store more documents (i increases if the size of the duplication region decreases). Figure 4 represents the above-described schematic at a particular instant of time. The documents are sorted by their LRU age and documents that are currently in cache 1 are shaded.

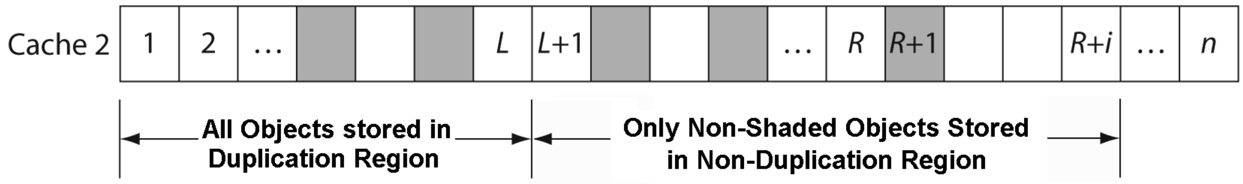
Cache 2
1
2
…
L
L+1
…
R
R+1
R+i
…
n
All Objects stored in Duplication Region
Only Non-Shaded Objects Stored in Non-Duplication Region

Figure 4: Objects Stored in Duplication and Non-Duplication Regions of Cache 2

The probability of a request for a document is given by its mean arrival rate divided by the total request arrival rate at the cache, i.e., $\text{Prob(request)} = \theta(x_k)/H_0$. The latency experienced by a user of cache 2 for a given sorted arrangement of documents is:

$$\left(\sum_{k=0}^{L_2}\frac{\theta(x_k)}{H_0}\cdot 0+\sum_{k=L_2+1}^{R_2+i-1}\frac{\theta(x_k)}{H_0}\big(\varphi_k\cdot D+(1-\varphi_k)\cdot 0\big)+\sum_{k=R_2+i+1}^{n}\frac{\theta(x_k)}{H_0}\big(\varphi_k\cdot D+(1-\varphi_k)\big)\right) \qquad (4)$$

For the first L_2 documents, all requests are satisfied by the cache and hence the delay is 0. For documents (L_2 + 1) through (R_2 + i – 1), the document may be in cache 2 or in cache 1. If the document is in cache 1 ($\varphi_k = 1$), then the document can be fetched from cache 1 with a delay of D. If the document is not tagged ($\varphi_k = 0$), then the document is returned from the local cache with no delay. Document (R_2 + i) is in cache 2 and hence incurs no delay. Finally, documents (R_2 + i + 1) through n may either be fetched from cache 1 if they are tagged ($\varphi_k = 1$) for a delay of D or fetched from the origin server ($\varphi_k = 0$) for a delay of 1. Without loss of generality, the server delay is normalized to 1. If the inter-cache latency were greater than server delay, then cooperating does not make sense, implying that $D \in [0,1]$. The expected latency for a user of cache 2 is given by averaging equation 4 over all possible values of document ages { $x_1 \in [0,\infty], x_2 \in [x_1,\infty],..., x_n \in [x_{n-1},\infty]$ } and all possible combinations of {i,j}.

$$EL_2 = \sum_{j=0}^{\min(L_2,R_1)}\sum_{i=0}^{R_1-j} p(j,i)\boldsymbol{E}_{\varphi|j,i}\cdot n!\int_0^\infty f(x_1)dx_1\int_{x_1}^\infty f(x_2)dx_2\cdots\int_{x_{n-1}}^\infty f(x_n)dx_n\left(\sum_{k=L_2+1}^{R_2+i-1}\frac{\theta(x_k)}{H_0}\varphi_k\cdot D+\right.$$
$$\left.\sum_{k=R_2+i+1}^{n}\frac{\theta(x_k)}{H_0}\big(\varphi_k\cdot D+(1-\varphi_k)\big)\right) \qquad (5)$$

EL_2 denotes the expected latency at cache 2. p(j, i) is the probability that there are j tagged documents among the first L_2 documents and i tagged documents before the (R_2 – L_2)th untagged document is encountered for the no-duplication region. We also define an operator, $\boldsymbol{E}_{\varphi|j,i}$, which allows us to consider all possible combinations in which documents can be tagged for a given *j* and *i*.

The decision problem faced by the cache operator is to select the value of L_2 that minimizes the expected latency. In the proceeding analysis, we investigate the solution of this decision problem. In Section 3.1, we begin with a problem in which cache 2 makes a duplication decision given cache 1 uses a traditional LRU policy ($L_1 = R_1$). This can refer to a scenario in which cache 1 is not strategic about optimal duplication levels or one in which cache 2 considers the worst case outcome wherein cache 1 makes no effort to reduce duplication in the array.[7] In Section 3.2, we optimize the duplication levels at both caches. We consider both decentralized and centralized decision contexts. We summarize our notation in Table 2.

n	Total number of documents on the web
R_i	Size of cache *i* (expressed as number of documents the cache can store)
L_i	Size of “duplication region” (also in terms of number of documents)
D	Delay incurred in fetching a document from other cache, $D < 1$
x_k	LRU Age (time since last request) for document *k*
$\theta(x_k)$	Instantaneous rate of access for a document with LRU age x_k. $\theta(x_k) = 1/(\alpha \cdot x_k + \beta); \quad \alpha, \beta > 0; \alpha < 1$
$f(x_k)$	Probability density of a document with LRU age x_k
H_0	The total mean access rate at the cache
φ_k	Indicator variable set to 1 if document *k* is currently in cache 1 or $\varphi_k = 0$ otherwise
p(*j*, *i*)	Probability that there are *j* tagged documents in the duplication region and *i* tagged documents before the ($R_2 - L_2$)*th* untagged document is encountered for the no-duplication region
EL_i	Expected latency at cache *i*

Table 2: Glossary of Terms

3.1 Optimal Duplication under LRU at Cache 1

In this Section, we assume that cache 1 uses a regular LRU policy. Cache 2 breaks up its cache into duplication and no-duplication regions. Under these assumptions, the probability *p*(*j*, *i*) is

$$p(j,i) = \binom{L_2}{j}\binom{R_2 - L_2 + i - 1}{i}\binom{n - R_2 - i}{R_1 - j - i} \cdot \binom{n}{R_1}^{-1} \qquad (6)$$

[7] ICP, the most commonly used protocol for cache cooperation, does not monitor duplication. Hence L_i=R_i for ICP.

where $j \in \{0, \cdots, L_2\}$; and $i \in \{0, \cdots, R_1 - j\}$. In the above expression, the number of ways in which we can tag *j* documents in the duplication region is $\begin{pmatrix} L_2 \\ j \end{pmatrix}$; *i* documents between (L_2 + 1)*th* and (R_2 + *i* – 1)*th* is $\begin{pmatrix} R_2 - L_2 + i - 1 \\ i \end{pmatrix}$; and ($R_1$ – *j* – *i*) documents between ($R_2 + i + 1$)*th* and *n-th* is $\begin{pmatrix} n - R_2 - i \\ R_1 - j - i \end{pmatrix}$ (see Figure 4). The total number of ways to tag R_1 documents in the sorted list of *n* documents is $\begin{pmatrix} n \\ R_1 \end{pmatrix}$.

The constraints on the binary indicator variables are $\sum_{k=L_2+1}^{R_2+i-1} \varphi_k = i$, and $\sum_{k=R_2+i+1}^{n} \varphi_k = R_1 - j - i$. In Appendix A, we have explicitly evaluated Equation (5) under these assumptions to arrive at the following expression for the expected delay at cache 2,

$$EL_2 = \frac{R_1}{n} \cdot D \cdot w(n, L_2) + \sum_{j=0}^{L_2} \sum_{i=0}^{R_1 - j} p(j, i) \frac{n - R_2 - R_1 + j}{n - R_2 - i} w(n, R_2 + i) \tag{7}$$

where $w(x, y) = \left(1 - \frac{y}{x}\right)^{\frac{1}{1-\alpha}}$. When *R* / *n* is small, we can approximate the above expression as shown in Appendix A,

$$EL_2 = \frac{R_1}{n} w(n, L_2) \cdot D + \left(1 - \frac{R_1}{n}\right) w\left(n, R_2 + R_1 \cdot \frac{R_2 - L_2}{n - R_1}\right) \tag{8}$$

The expected delay may be interpreted as EL_2 = Pr(request satisfied locally)·0 + Pr(request satisfied by cache 1)·*D* + Pr(request satisfied by origin server) ·1. Note that $\frac{R_1}{n} w(n, L_2)$ is decreasing in L_2 and $\left(1 - \frac{R_1}{n}\right) w\left(n, R_2 + R_1 \cdot \frac{R_2 - L_2}{n - R_1}\right)$ is increasing in L_2. This observation highlights the main tradeoff in choosing the size of duplication region (L_2): if the cache manager increases L_2, more requests get satisfied locally, fewer requests are satisfied at cache 1 and a greater proportion of requests go to the origin server. In other words, the local hit rate increases but array hit rate decreases. ICP and Summary

Cache attempt to maximize the local hit rate and set $L_2= R_2$. CARP attempts to maximize the array hit rate and sets L_2=0. The expected latency depends on both the local and array hit rates.

PROPOSITION 1: Zero duplication is preferable to unmonitored duplication for $D<D_{Th}$ and vice-versa, where $D_{Th} = \left(\frac{n}{R_1} - 1\right)\left[\frac{\left(1-\frac{R_2}{n}\right)^{\frac{1}{1-\alpha}} - \left(1-\frac{R_2}{n-R_1}\right)^{\frac{1}{1-\alpha}}}{1-\left(1-\frac{R_2}{n}\right)^{\frac{1}{1-\alpha}}}\right]$.

The proof is in Appendix B. Given most proxy servers support both ICP and CARP and let the cache operator select the duplication level, Proposition 1 provides an important qualitative insight to guide the decision. Specifically, zero duplication is preferable to unmonitored duplication when inter-cache latency (D) or request locality (α) are low. The finding can be explained as follows. At high inter-cache latency, there are limited gains from fetching an object from another cache in the event of a local miss. So the emphasis is on maximizing local hit rate through unmonitored duplication. In contrast, at low inter-cache latency, an array hit is almost as effective as a hit from the local cache. So the emphasis is on maximizing the array hit rate through zero duplication. Similarly, when request locality is very high, the locality is best exploited by ensuring that recently requested objects are stored locally as opposed to eliminating them if they exist in another cache. Thus, unmonitored duplication is preferred at high locality. The converse is true at low locality.

The optimal duplication level corresponds to the value of L_2 that minimizes the expected latency. That is, $L_2^* = \min_{L_2}\{EL_2\}$. The objective function is globally convex in L_2. The minimization problem has a closed form solution given by

$$L_2^* = n - \frac{n\cdot(n-R_2)}{(n-R_1)\cdot D^{\frac{1}{\alpha}-1} + R_1} \qquad (9)$$

(9) is always less than the cache size R_2. However, the expression may sometimes result in a negative value. Incorporating the non-negativity constraint, the proposition follows:

PROPOSITION 2: *The optimal size of the duplication region is given by*

$$L_2^* = \max\left(0, n - \frac{n \cdot (n - R_2)}{(n - R_1) \cdot D^{\alpha^{-1}-1} + R_1}\right) \tag{10}$$

COROLLARY 1: *The optimal size of the duplication region is (a) non-decreasing in the inter-cache wait time, D, (b) non-decreasing in request locality, (c) non-decreasing in the size of the other cache R_1, and (d) non-decreasing in cache size R_2.*

The proofs are in the Appendix B. In Figure 5, we illustrate results (a) and (b) from Corollary 1 for a case in which {n = 5000, R_1 = 500, R_2 = 400}. When D = 0, i.e. there is no perceivable difference between fetching the document from the local cache or the other cache, then L_2^* = 0. That is, there will be no duplication of content and a scheme like CARP is desirable. For small inter-cache latencies, zero duplication continues to perform well. However, as the inter-cache latency increases, it is optimal for the cache manager to allow some duplication in order to ensure that the most popular documents are locally stored irrespective of their presence in the other cache.[8]

Now consider the impact of request locality. Recall that a high value of α implies that LRU age is a good predictor of future requests. Thus, as α increases, it becomes appealing to always store the items with the lowest LRU age. Eliminating a recently requested document just because it exists in the other cache can result in significant decline in local hit rates which in turn drives up the latency. As a result, we observe that (L_2^* / R_2) weakly increases with α .

Finally, consider the impact of the cache sizes. If the size of either cache increases, the overall capacity of the array also increases. The additional capacity in the array reduces the negative impact of duplication resulting in an increase in L_2. A clear recommendation from Corollary 1 is that policies with zero or low duplication of content (CARP, Two-exit LRU with small duplication, etc) are recommended when the caches are located close by, cache capacity is limited and temporal locality in requests is

[8] In our scheme, $L_2^* / R_2 = 1$ only when D = 1 because there is no cost of querying the other cache (an index as in summary cache is maintained and is always current). If no index is maintained or if the index is not current, then $L_2^* / R_2 = 1$ at high inter-cache latencies that are less than the origin server delay.

relatively low (all else constant). When caches are geographically farther apart, temporal locality is high and the cache capacities are high, then greater levels of duplication is preferred.

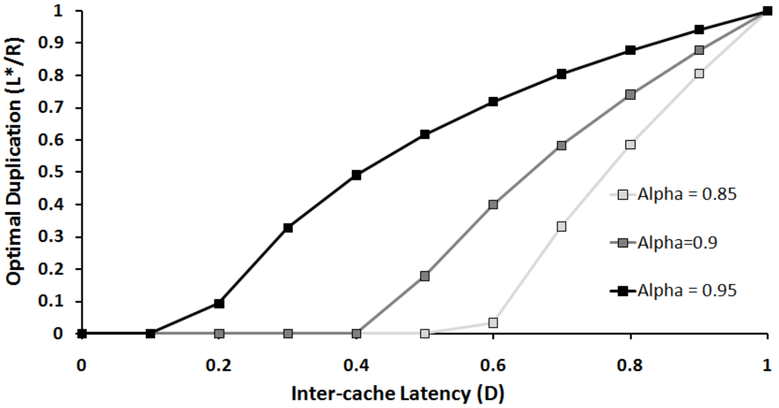

Figure 5: Impact of Inter-cache Latency and Request Locality on Optimal Duplication

COROLLARY 2: Unmonitored duplication ($L_i= R_i$) is optimal if D=1 or $\alpha = 1$, and no duplication (L_i=0) is optimal if $D^{\alpha^{-1}-1} \leq 1 - \frac{R_2}{n-R_1}$.

Corollary 2 indicates that unmonitored duplication (e.g. ICP, Summary Cache) is optimal if the inter-cache latency or request locality is high. Conversely, zero duplication (e.g. CARP) is optimal when inter-cache latency and request locality are low. This is consistent with Proposition 1. Proposition 1 identified the conditions under which zero duplication is better than unmonitored duplication and vice-versa. In contrast, Corollary 2 identifies the conditions under which these decisions are optimal. In short, Proposition 1 informs a cache operator considering only zero or unmonitored duplication whereas Corollary 2 informs an operator willing to expend the effort to logically partition the caches to achieve any level of duplication. Figure 6 plots the expected latency at cache 2 with zero duplication (L_2 = 0), unmonitored duplication ($L_2 = R_2$) and optimal duplication ($L_2 = L_2^*$). The remaining parameters are {n = 5000, R_1 = 500, R_2 = 400, α =0.95}. It is clear that there exist a wide range of parameters under which it is optimal for the cache operator to select unmonitored duplication or zero duplication. At the same time, choosing the wrong duplication level (e.g., choosing unmonitored duplication when no duplication is optimal), can result in significant deterioration of performance. The expected latency with optimal duplication traces the inner envelope of the latency curves of unmonitored and zero duplication and provides the maximum benefit at intermediate levels of inter-cache latencies.

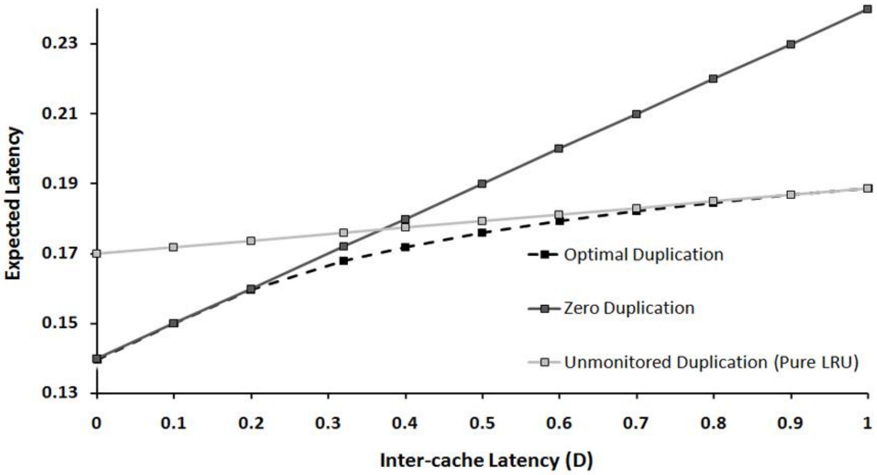

Figure 6: Average Latency with Different Duplication Schemes

Our results highlight how various factors including inter-cache latency and request locality affect the optimal duplication levels. We now turn to the optimization of duplication levels at both caches.

3.2. Optimizing Duplication at Both Caches

We evaluate the problem under two decision scenarios. The first is a setting in which the duplication decisions are decentralized, i.e. made independently at each cache. Whenever cooperative caching spans organizational boundaries, it is expected that individual organizations choose policies that are locally optimal (example, minimize latency for their own users). For example, this corresponds to a situation in which two ISPs may peer with one another at bilateral peering points or at exchanges such as Packet Clearing House and Equinix. Cooperative caching implementations in the public domain, such as IRCache and w3cache, are also relevant here. The second decision setting is a centralized one, which corresponds to a situation in which an ISP or a large organization implements cooperative caching across its proxy servers.

3.2.1. Decentralized Coordination

When both caches are strategic, we need to compute each cache's best response to the other cache's decision in order to compute the equilibrium outcomes. The expression for expected latency is the same as equation 5. However, *p*(*j*, *i*), the probability that there are *j* tagged documents among the first L_2 documents and *i* tagged documents before the $(R_2–L_2)th$ untagged document is encountered for the no-duplication region, has a different functional form. This is because the number of documents collectively stored increases as cache 1 reduces the size of its duplication region L_1. Under this game structure, *p*(*j*,*i*) can be computed as shown in Appendix C,

$$p(j,i)=\binom{L_2}{j}\binom{R_2-L_2+i-1}{i}\binom{n-R_2-i}{R_1-j-i}\binom{R_1-j}{L_1-j}\cdot\binom{n-L_2}{R_1-L_1}^{-1}\binom{n+L_1-R_1}{L_1}^{-1} \qquad (11)$$

Combining Equations 11 and 5 and simplifying as shown in Appendix C,

$$EL_2=\frac{1}{n-L_2}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right)\cdot D\cdot w(n,L_2)+\sum_{j=0}^{L_2}\sum_{i=0}^{R_1-j}p(j,i)\frac{n-R_2-R_1+j}{n-R_2-i}w(n,R_2+i) \qquad (12)$$

where $w(x,y)=\left(1-(y/x)\right)^{\frac{1}{1-\alpha}}$.

As before, we can interpret (12) as the probability that the request goes to cache 1 times *D*, plus the probability that the request goes to the origin server times 1. Assuming that R_1/n and R_2/n are small, we further show that (12) can be approximated by,

$$\begin{aligned}EL_2=&\frac{1}{n-L_2}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right)\cdot D\cdot w(n,L_2)\\&+\left(1-\frac{1}{n-L_2}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right)\right)\cdot w\left(n,R_2+\frac{R_1(n+L_1-R_1)-L_1L_2}{n+L_1-L_2-R_1}\frac{R_2-L_2}{n-R_1}\right)\end{aligned} \qquad (13)$$

The response function for cache 2 can be computed by $L_2^{*}(L_1)=\min_{L_2}\{EL_2\}$. Similarly, the response function for cache 1 is obtained by setting up the expression for the expected latency at cache 1, EL_1, and solving $L_1^{*}(L_2)=\min_{L_1}\{EL_1\}$. Unfortunately these decision problems have no closed form solutions. But the following property of the response functions can be derived,

PROPOSITION 3: *The best response curves,* $L_2^{*}(L_1)$ *and* $L_1^{*}(L_2)$, *are non-increasing in* L_1 *and* L_2 *respectively. Further, the existence of an equilibrium is guaranteed.*

The proof is in Appendix C. The response functions are non-increasing. Thus, this is a game of strategic substitutes wherein a reduction of the duplication region at one cache results in an expansion of the duplication region at the other.[9] Proposition 3 highlights a key result regarding the nature of the strategic interaction between caches. If a cache allocates more resources towards eliminating duplicate documents, this creates an incentive for the other cache to free-ride and increase the size of its own duplication region. This can be explained as follows. A small increase in the size of the duplication region at a cache (say cache 2) provides a benefit in the form a slight increase in the local hit rate but also imposes a cost in the form of a slight increase in requests forwarded to the origin server (due to a drop in the array hit rate). However, the latter cost is relatively small when the other cache (cache 1) reduces the size of its duplication region. This allows cache 2 to reduce the size of its duplication region without having to deal with a significant increase in requests forwarded to the original server. As a result, it is hard to simultaneously induce both caches to contribute greater resources towards reducing duplicate documents in the system in the absence of additional incentives such as payments.

While there is no closed form solution for the equilibrium values, they can be evaluated numerically.[10] Figure 7 plots the response curves of the two caches when {n = 5000, R_1 = 500, R_2 = 400, $\alpha = 0.9$ and D = 0.5}. The equilibrium solution is given by { $L_1^* = 197, L_2^* = 92$ }. The equilibrium L_2^* is higher than would be the case if cache 1 uses a regular LRU policy (L_2^* = 71 when L_1=500). The fact that cache 2 is aware that cache 1 has an incentive to reduce L_1 results in an increase in L_2.

[9] If a boundary condition has been reached (L_2=0 or L_2= R_2), then the duplication level may remain unchanged resulting in the response functions being non-increasing rather than strictly decreasing.

[10] However, multiple equilibria can exist.

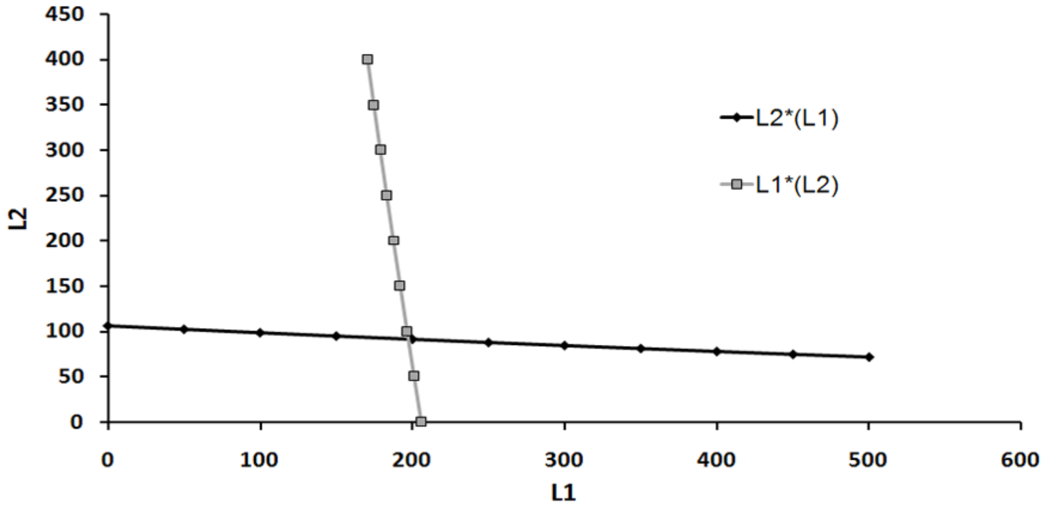

Figure 7: Best Response Curves and Equilibrium Duplication Levels for the Two Caches

PROPOSITION 4: *The best response curves, $L_2^*(L_1)$ and $L_1^*(L_2)$, are both non-decreasing in inter-cache latency, D.*

Holding the other player's choice of *L* fixed, the size of the duplication region at a cache weakly increases with inter-cache latency. This does not imply that equilibrium duplication levels are non-decreasing in *D*. In Figure 8, we illustrate that the equilibrium duplication levels can decrease with *D* even though the response curves $L_2^*(L_1)$ and $L_1^*(L_2)$ are monotonically non-decreasing in *D*. This result can be explained as follows. A decrease in *D* may cause one cache to lower the size of its duplication region, *L*. However, since the cache's decisions are strategic substitutes, there is a second-order effect wherein a decrease in the size of the duplication region by one cache encourages the other to increase *L* resulting in the non-monotonicity in equilibrium.

Finally, Figure 9 plots the average latency at cache 2 under the equilibrium and contrasts it with the average latency when both caches choose zero duplication and both choose unmonitored duplication. The remaining parameters are {$n = 5000$, $R_1 = 500$, $R_2 = 400$, $\alpha = 0.95$}. As before, choosing unmonitored duplication when zero duplication is optimal (and vice-versa) can result in significant performance deterioration. Also, the average latency under the equilibrium again traces the inner envelope of the latency curves of unmonitored and zero duplication and provides the maximum benefit at intermediate levels of inter-cache latencies. These results are qualitatively similar to Figure 6 and they help underscore the fact that tuning the duplication levels can provide significant gains even under strategic behavior.

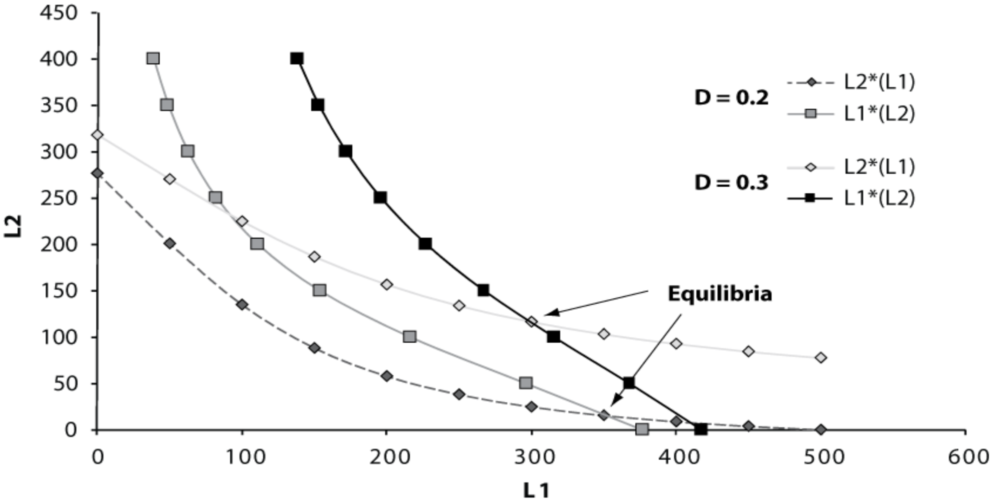

Figure 8: Monotonic Response Functions and Non-monotonic Equilibria

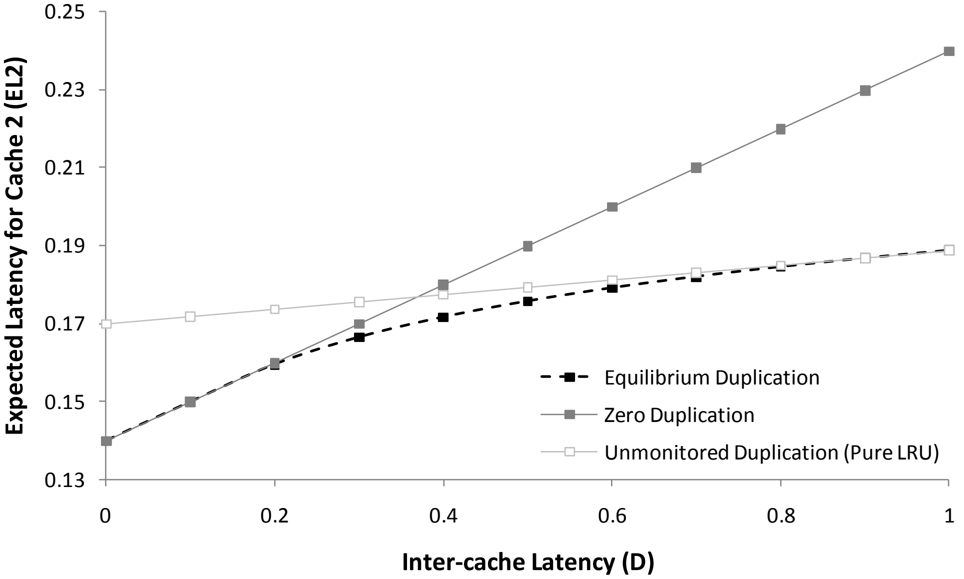

Figure 9: Average Latency with Different Duplication Schemes

We now explore the optimal duplication levels chosen by a central planner and contrast the solution with the equilibrium duplication levels computed above.

3.2.2. Centralized Coordination

Consider a central planner that sets the optimal duplication level at each cache in order to minimize overall latency in the system (i.e., maximize social welfare). The decision problem can be stated as:

$$\min_{L_1,L_2}\left(\frac{H_1 \cdot EL_1 + H_2 \cdot EL_2}{H_1 + H_2}\right) \tag{14}$$

where H_1 and H_2 are the request arrival rates at caches 1 and 2 respectively, and are specified by equation 3. The objective function represents the average latency experienced in the system, which is a weighted sum of the average latencies at the individual caches. Assuming the same values for α and β at the two caches, the decision problem is $\min_{L_1,L_2}(EL_1 + EL_2)/2$.

In Table 3, we present the solution for different values of inter-cache latency and request locality. The remaining parameters are {n = 5000, R_1 = 500, R_2 = 400}. For moderate values of inter-cache latency, we observe that the centralized solution entails greater resource commitment from cache 2 (i.e. low L_2) and lower commitment from cache 1 (high L_1). Setting L to a very low value at one cache ensures minimal duplication of documents across the two caches. The central planner can then set a high value of L at the other cache (i.e., an LRU-like policy) to ensure low latency at that cache. Thus the social optimum may entail penalizing users at one cache in order to ensure very low average latency at the other. When cooperative caching is done by independent organizations, it is not clear whether solutions with such asymmetric resource commitment can be implemented in the absence of additional mechanisms such as payments.[11]

For each of the parameter values in Table 3, we also computed the expected latency under the centralized and decentralized setups. The expected latency under the centralized setup is 0.00-3.00% lower than that achieved under decentralized decision making. As expected, centralized decision-making is better.

D	α = 0.85		α = 0.90		α = 0.95	
	L_1^*	L_2^*	L_1^*	L_2^*	L_1^*	L_2^*
0	0	0	0	0	0	0
0.1	0	0	0	0	0.01	0
0.2	0	0	0	0	150.65	45.95

[11] At low to medium levels of request locality, we sometimes observe asymmetric solutions even when caches are symmetric. We provide some examples in the online appendix.

0.3	0	0	0	0	233.70	133.98
0.4	0	0	106.57	0	294.85	196.89
0.5	51.12	0	190	80.43	343.38	245.95
0.6	162.21	33.39	263.10	160.50	383.71	286.20
0.7	244.81	135.15	329.76	230.28	418.28	320.38
0.8	330.27	228.7	390.85	292.37	448.58	350.09
0.9	415.52	316.44	447.32	348.52	475.59	376.39
1	500	400	500	400	500	400

Table 3: Socially Optimal Duplication Levels (R_1 = 500, R_2=400)

3.3. Discussion

Our analysis yields a number of insights for cache operators. First we find that even if other caches do not do much to monitor duplication, a selfish cache may still want to take steps to control duplication. Thus free-riding (setting $L=R$) need not be optimal even if other caches do not attempt to eliminate duplicate documents. The primary tradeoff associated with duplication is as follows: if the cache operator does not monitor duplication and chooses a placement/replacement policy that is locally optimal, then it maximizes the local hit rate but also results in fewer requests being satisfied at other caches. At the same time, eliminating documents that are currently stored elsewhere has the risk of lowering local hit rate. As a result, the cache operator may benefit from controlling duplication even when other caches take no such action.

We also find that there exist a range of settings in which it might just suffice to choose one of two extreme solutions. Specifically, zero duplication is optimal at low inter-cache latencies or low levels of request locality. And conversely, unmonitored duplication is optimal at high inter-cache latencies or high levels of request locality. Choosing zero (unmonitored) duplication when unmonitored (zero) duplication is optimal can have an adverse impact and negate much of the value from cache cooperation. This observation is particularly important because most proxy servers including Squid and Sun Java Proxy Server support both zero and unmonitored duplication and leave the specific choice to the cache operator. Cache operators must be careful in selecting the right setting. Finally, at intermediate values of request locality and inter-cache latency, the extra effort of implementing two-exit policies may be justified. The size of the duplication region in these two exit policies is non-decreasing in request locality, inter-cache latency and cache sizes.

Cooperative caching can be implemented within large organizations as with an ISP with multiple proxy servers or across organizations as with bilateral peering among ISPs and cache peering at exchanges like Equinix. A key question here relates to the impact of strategic behavior on the success of cache peering. Our analysis provides a few insights here as well. First, we find that the interaction among strategic caches is a game of strategic substitutes. That is, it is best for a cache to employ lesser resources towards eliminating duplicate documents when other caches employ more resources towards reducing document duplication. Thus it may be hard to simultaneously induce multiple caches to contribute towards the best global performance in the absence of mechanisms such as payments and audits. We also find, as expected, that centralized decision-making does better than decentralized decision-making. More importantly, the size of the duplication regions under this centralized setup can be highly asymmetric. This in turn can benefit users connected to the cache with the largest duplication region while penalizing users who are connected to the cache that has the smallest duplication region. This has the potential to be perceived as unfair by one set of users. Thus, issues of fairness may need to be addressed when implementing the optimal centralized decision.

All the above insights are derived from a model in which we make several simplifying assumptions for tractability. We now test the robustness of these insights by relaxing several of the key assumptions made in this section.

4. Robustness of Results

We approach robustness tests in two parts. First, we test the robustness of the analytical findings from Section 3.1 by relaxing two key independence assumptions in our two-cache setup: (i) inter-cache latency is independent of traffic, (ii) requests are independent at the two caches. These assumptions are relaxed within the framework of our analytical model. Next, we evaluate the findings tied to centralized versus decentralized coordination in a setting with more than two caches and with real-world request traces. These tests are conducted within a simulation environment.

4.1. Extending Analytical Model to Relax Independence Assumptions

4.1.1. Traffic-dependent Inter-cache Delay

In Section 3, the inter-cache latency is assumed to be independent of the traffic between the two caches. However, traffic can influence the waiting and processing times for requests at a cache and thereby affect the overall latency. We now explicitly model the waiting/processing time at a cache as a function of the traffic.

In Equation (8), $\frac{R_1}{n} w(n, L_2)$ is the probability that documents are fetched from cache 1. Since $\frac{1-\alpha}{\beta} n$ is the total demand, $\lambda = \frac{1-\alpha}{\beta} n \cdot \frac{R_1}{n} w(n, L_2)$ is the mean traffic intensity to cache 1. If τ denotes the mean service time at cache 1 (i.e., service rate is $\mu = 1/\tau$), then the mean time in the system for requests forwarded to cache 1 is given by $\frac{1}{\mu - \lambda} = \frac{\tau}{1 - \left(\frac{1-\alpha}{\beta} n \frac{R_1}{n} w(n, L_2) \right) \tau}$. Thus, the expression for expected latency at cache 2 is,

$$EL_2 = \frac{R_1}{n} w(n, L_2) \left(D + \frac{\tau}{1 - \left(\frac{1-\alpha}{\beta} R_1 w(n, L_2) \right) \tau} \right) + \left(1 - \frac{R_1}{n} \right) w\left(n, R_2 + R_1 \cdot \frac{R_2 - L_2}{n - R_1} \right)$$

The cache operator's decision problem is, $L_2^* = \min_{L_2} \{EL_2\}$. There is no closed-form solution for L_2^*. However, applying the conjugate pairs theorem, we can show that:

PROPOSITION 5: *With traffic-dependent inter-cache latency, the optimal size of the duplication region is (a) non-decreasing in the inter-cache delay D, (b) non-decreasing in request locality α for τ below some positive threshold, (c) non-decreasing in cache size R_2, (d) non-decreasing in cache size R_1, and (e) non-decreasing in mean service time τ.*

The proof is in Appendix D. These results are consistent with the results of Section 3. However if τ is very high, the result regarding request locality can change signs as highlighted in the appendix. That the size of the duplication region is non-decreasing in the mean service time τ is expected because the value from cooperative caching reduces when the other proxy cache takes too long to process requests.

Consequently, the cache operator is better off ensuring that the requests for popular documents get serviced locally.

4.1.2. Correlated Demand

In Section 3.1, the expression for *p*(*j*,*i*) assumes that requests at the two caches are independent. Here, we relax the assumption to consider positively correlated requests. For arbitrary correlation structures, it is hard to derive an analytical expression for the expected latency so we consider a specific form below.

In the original model, the number of tagged documents (*j*) in cache 2's duplication region varies from zero to L_2 and the probability distribution of *j* is obtained by assuming requests are independent. If requests are positively correlated, we are more likely to see higher values of *j* than the lower values. Thus the probability density around low values of *j* reduces and that around high values of *j* increases. To model this, we truncate the parameter *j* by removing its small values. That is, *j* varies from min(*l*, L_2) to L2. A low value of *l* is close to our original independence assumption and a high value indicates that the popular items at cache 2 are likely to also be in cache 1. Specifically, we modify the probability *p*(*j*,*i*) as follows:

$$p_c(j,i) = \binom{L_2}{j}\binom{R_2 - L_2 + i - 1}{i}\binom{n - R_2 - i}{R_1 - j - i} \cdot NF^{-1},$$

Where $l \le j \le L_2$ and $0 \le i \le R_1 - j$. *NF* is a normalization factor such that $\sum_{j=l}^{L_2} \cdot \sum_{i=0}^{R_1 - j} p_c(j,i) = 1$.

Under this new model, the expected delay can be computed in the same manner as in Section 3. The new expression is:

$$EL_2 = D\frac{R_1 - \varphi}{n}\left(1 - \frac{L_2}{n}\right)^{\frac{\alpha}{1-\alpha}} + \left(1 - \frac{R_1 - \varphi}{n - L_2}\right)\left(1 - \frac{nR_2 - L_2\left(R_1 + R_2 - \varphi\right)}{n\left(n - L_2 - R_1 + \varphi\right)}\right)^{\frac{1}{1-\alpha}},$$

where, $\varphi = l + \frac{{}_3F_2\left(2, 1 + l - L_2, 1 + l - R_1; 2 + l, 2 + l + n - L_2 - R_1; 1\right)(l - L_2)(l - R_1)}{{}_3F_2\left(1, l - L_2, l - R_1; 1 + l, 1 + l + n - L_2 - R_1; 1\right)(l + 1)(1 + l + n - L_2 - R_1)}$, and ${}_aF_b$() denotes the HyperGeometric function.

The optimal value L_2^* that minimizes EL_2 can be obtained numerically. Figure 10 shows that the size of the duplication region is non-monotonic with the correlation parameter l ($\alpha = 0.95, D = 0.4$). Figure 11 shows that the size of the duplication region is non-decreasing in the inter-cache latency and request locality (n=5000, R_1=500, R_2=400, l=40). These results are consistent with those in Section 3.

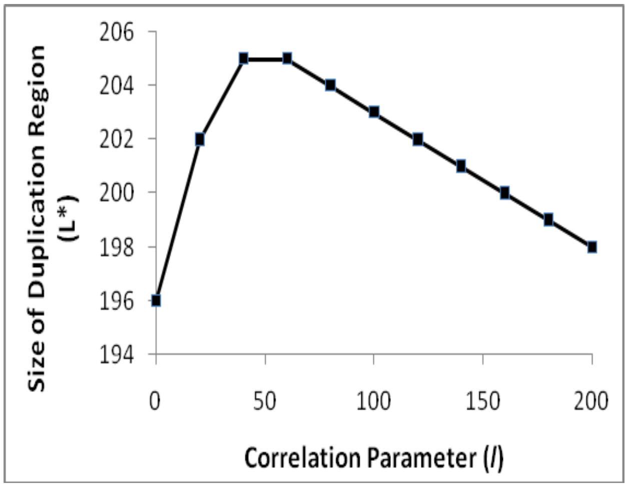

Figure 10: Impact of Correlated Requests on Optimal Duplication

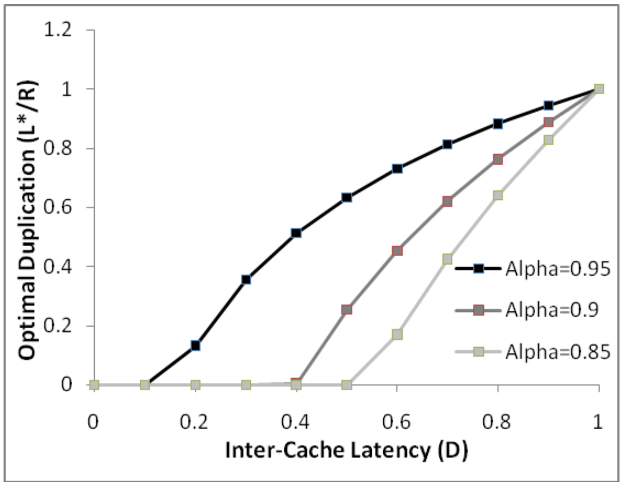

Figure 11: Impact of Inter-cache Latency and Locality on Optimal Duplication

4.2 Simulations

In this Section, we use trace-driven simulations to evaluate our findings under more realistic requests and with multiple caches. We used a large request stream collected on January 09, 2007 by the Urbana-Champaign proxy server of the IRCache project[12]. The trace consists of 423,968 requests for 207,206 unique URLs submitted by users of the proxy server. The log reports the size of the response as well as the response time for each request. For size, the trace records number of bytes written to the client rather than size of the data object. Because header sizes vary across requests, there can be minor variations in size across requests for the same URL. In accord with Kelly et al. (1999), we define the size of a URL as the maximum recorded transfer size among all requests for it and assume that the URLs remain unchanged throughout the day. The object sizes in our trace range from 0.2 KB to 129 MB with an average size of 26.71 KB. Because the trace does not report the origin server's response time during

[12] www.ircache.net

cache hits, we use the maximum recorded response time for a URL as the origin server's response time for that URL.

All requests in the trace are generated by users of a single proxy cache. In order to test cooperative caching schemes, we create M caches in any simulation run (M is an input parameter) and randomly assign the users in the log file to one of the M caches. Once users are assigned, requests at individual caches are generated in accordance with the sequence and timing specified in the trace. The inter-cache latency is fixed as a constant fraction, D, of the origin server's response time for a URL. All caches are initially empty and are populated in accordance with the two exit policy discussed in Section 3. The results from the first 25,000 requests at any cache are discarded to eliminate data from the initial period when caches are not yet full.

The input parameters for the simulation are the number of caches (M), the caches sizes (R_i) and the ratio D. Other parameters are determined by the trace. In the simulations, we considered the cases of two, four and five caches ($M \in \{2,4,5\}$) as well as a variety of cache sizes. Below, we report results for M=4 and asymmetric cache sizes (R_1=62.6 MB, R_2=83.4MB, R_3=104.4MB, R_4=125.22MB). The smallest cache is 2400 times the average size of an object in our trace and the largest cache is 4800 times the average size of an object. The results are qualitatively similar for other values. In contrast, our results are sensitive to the inter-cache latency. Hence, we report the results for a range of D.

Our simulations test four policies. The first two policies are L_i=0 and L_i=R_i. The third policy allows each cache to independently determine a locally optimal L_i and evaluates the performance at the resulting equilibrium. There is no simple method to compute the equilibrium in our multi-cache simulations. We consider an iterative best-response procedure. To determine the best response for cache i (i.e. optimal L_i given L_{-i}), we use simulations to compute the expected latency at cache i for all candidate values of L_i.[13] The optimal L_i minimizes the expected latency at cache i. Given a starting point L^0 = (L_1, L_2,...,L_M), the iterative best-response procedure sequentially optimizes the duplication levels of caches 1

[13] We restricted the candidate values of L_i to {0, 0.01R_i, 0.02R_i, … , R_i}.

through M and repeats the process until the duplication levels converge. In order to account for the possibility of multiple equilibria, we consider several randomly chosen starting points L^0. The fourth policy we consider is the socially optimal duplication level. We exhaustively consider all possible values of $(L_1, L_2,...,L_M)$ and choose the one that minimizes the average latency in the cache array.[14]

In Table 4, we report the average latency in the cache array under each of the four policies. The average latency under zero duplication and unmonitored duplication are in the first two columns. The column labeled "Equilib." presents the average latency realized in equilibrium with decentralized decision-making. The socially optimal policy is labeled "Min". We additionally report the maximum expected latency observed during our exhaustive evaluation of all possible $(L_1, L_2,...,L_M)$. This maximum latency along with the socially optimal value help benchmark the other three policies. For consistency with Section 3, the expected latency under a regime where all requests are forwarded to the origin server is normalized to 1. In accord with the results in Section 3, we find that policies with zero duplication ($L_i = 0$) are desirable at low inter-cache latency and policies with unmonitored duplication ($L_i = R_i$) are desirable at high latencies and that optimizing duplication levels provides benefits over both policies. As expected, there are costs to decentralized decision-making although they are somewhat low. Finally, we investigated the shape of the best response function by comparing the best response at each cache obtained during the iterative procedure. Specifically, we did a pairwise comparison of all computed $L_i^*\left(\sum_{j=1..M,\, j\neq i} L_j\right)$ at each cache. 86.33% of these pairwise comparisons were non-decreasing providing support for the observation that this is a game of strategic substitutes.

	Average Latency				
	$L_i = 0$	$L_i = R_i$	Equilib.	Min	Max
D=0.0	0.568	0.590	0.568	0.568	0.590
D=0.2	0.606	0.597	0.585	0.582	0.606
D=0.4	0.644	0.606	0.598	0.595	0.644
D=0.6	0.683	0.614	0.611	0.610	0.683
D=0.8	0.721	0.623	0.623	0.623	0.721

[14] Simulations were run in parallel on a grid computing system in order to address the computational overhead imposed by an exhaustive evaluation.

D=1.0	0.759	0.635	0.635	0.635	0.759

Table 4: Expected Latency from Different Duplication Policies

In summary, we find that the key insights from the analytical model continue to hold in the trace-driven simulations.

5. Conclusions

Tuning the level of duplication across cooperating caches is a key operational decision for cache operators. They may choose to not monitor duplication (e.g., by choosing ICP, Summary Cache or Cache Digest), have zero duplication across caches (e.g. by choosing CARP), or expend additional effort to achieve intermediate levels of duplication (e.g., HOME, two-exit LRU). The optimal decision depends on the deployment context and operators have to experiment considerably before making the decision. We develop a model to study the relevant tradeoffs and generate insights regarding optimal duplication in strategic and non-strategic settings.

Our study represents an initial foray into understanding the tradeoffs tied to document duplication. However, the study has several limitations. First, although our study provides several insights regarding optimal duplication levels, protocol selection in cooperative caching requires additional considerations. One of these relates to overheads associated with the different protocols. As mentioned earlier, ICP queries all caches upon a local miss and thus generates significant query traffic when the cache array is large. Thus, ICP is undesirable for large cache arrays. Further, Summary cache and Cache Digest require use of directories. This imposes overheads tied to maintaining the directories and also storing them. Fortunately, these issues have received considerable attention. For example, studies show that the directories can be maintained with low overheads if caches delay propagation of directory updates. Further, directories can be stored in a compact manner by using Bloom filters. URL hashing used in CARP, Two-exit LRU and Home do not have these communication and storage overheads, with the latter two protocols capable of implementing any level of duplication. However, known challenges with URL hashing are the significant disruption every time a cache is added or removed and achieving even load distribution among proxy caches. Recent work on consistent hashing proposes several solutions to these hash disruption issues. These considerations need to be factored into protocol selection. Future work can look at protocol selection more holistically after accounting for all relevant factors including duplication, overheads and implementation complexity.

Another limitation of our work is that we assume that proxy caches are always closer than the origin server. In global-scale caches, this is not always true and a key challenge is that of proxy pruning, namely selecting between remote proxies and origin server. Our model does not account for this. We also do not consider cache hierarchies and instead assume all caches are at the same level. Incorporating proxy pruning and cache hierarchies into our framework is clearly of value. Finally, a promising direction for future work is the study of mechanisms to ensure that the socially optimal duplication levels can be attained in practice under decentralized decision-making. Such a study can provide prescriptions for contracting and auditing in cooperative caching.

References

Bose, I., H. K. Cheng. Performance Models of a Firm's Proxy Cache Server. *Decision Support Systems*, Vol. 29, Issue 1, July 2000, Pages 47-57.

Chan, Y. M., J. Womer, J. K. MacKie-Mason, S. Jamin. One size doesn’t fit all: Improving network QoS through preference driven web caching. Proc. 27th Annual Telecomm. Policy Res. Conf., Alexandria, VA, 1999.

Che, H., Y. Tung, and Z. Wang, "Hierarchical web caching systems: Modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, Sept. 2002.

Chiang, I. R., P. Goes, Z. Zhang. Periodic Cache Replacement Policy for Dynamic Content at Application Server. *Decision Support Systems*, Vol. 43, Issue 2, March 2007, Pages 336-348.

Chuang, J., M. Sirbu. Distributed network storage with quality-of-service guarantees. *J. Network Comput. Appl.*, 23(3) 163–185, 2000.

Datta, A., K. Datta, H. Thomas, D. VanderMeer, “WORLD WIDE WAIT: A Study of Internet Scalability and Cache-Based Approaches to Alleviate it,” *Management Science*, Vol. 49, No. 10, October 2003.

Debreu, D. 1952. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, Vol.38, 886-893.

Dogan, K., C. Kaya and V. Mookerjee, “An Economic and Operational Analysis of the Market for Content Distribution Services”, Proceedings of the International Conference on Information Systems, December 14-17, 2003, Seattle, WA.

Du, A., X. Geng, R. Gopal, R. Ramesh, A. B. Whinston. Capacity Provision Networks: Foundations of Markets for Sharable Resources in Distributed Computational Economies. *Information Systems Research*, Vol. 19, No. 2, June 2008, pp. 144-160.

Dutta, K., S. Soni, S. Narasimhan, and A. Datta. Optimization in Object Caching. *INFORMS Journal on Computing*, Vol. 18, No. 2, Spring 2006, pp. 243-254.

Fan, L., P. Cao, J. Almeida, and A. Z. Broder. Summary Cache: a scalable wide-area Web cache sharing protocol. Proceedings of Sigcomm, 1998.

Fang, X., O. Sheng, W. Gao, B. Iyer. A Data-Mining-Based Prefetching Approach to Caching for Network Storage Systems. *INFORMS Journal on Computing*, Vol. 18, No. 2, Spring 2006, pp. 267-282.

Geng, X., Gopal, R. D., Ramesh, R., and Whinston, A. B., “Scaling Web Services with Capacity Provision Networks,” *IEEE Computer*, Vol. 36, No. 11, November 2003.

Gualteiri, M. and J. Staten. Best Practices: Attaining and Maintaining Blazing Fast Web Site Performance. Forrester Industry Report. February 2009.

Hosanagar, K., J. Chuang, R. Krishnan, M. Smith. Service Adoption and Pricing of Content Delivery Network (CDN) Services. *Management Science*, Vol. 54, No. 09, September 2008.

Hosanagar, K. and Y. Tan. Optimal Duplication in Cooperative Web Caching. Proceedings of the Fourteenth Annual Workshop on Information Technologies and Systems (WITS), 92–97, December 2004.

Hosanagar, K., R. Krishnan, J. Chuang, and V. Choudhary. Pricing and Resource Allocation in Caching Networks with Multiple Levels of QoS. *Management Science*, 51 (12), 2005.

Kaya, C., G. Zhang, Y. Tan, V. Mookerjee. An admission-control technique for delay reduction in proxy caching. *Decision Support Systems*, Vol. 46, Issue 2, January 2009, Pages 594-603.

Kelly, T., S. Jamin, J. K. MacKie-Mason. Variable QoS from shared web caches: User-centered design and value-sensitive replacement. MIT Workshop on Internet Service Quality Economics, Cambridge, MA, 1999.

Kumar, C. Performance evaluation for implementations of a network of proxy caches. *Decision Support Systems*, Vol. 46, Issue 2, January 2009, Pages 492-500.

Kumar, C., J. Norris. A new approach for a proxy-level web caching mechanism. *Decision Support Systems*, Vol. 46, Issue 1, December 2008, Pages 52-60.

Mookerjee, V. and Y. Tan, “Analysis of a Least Recently Used Cache Management Policy for Web Browsers,” *Operations Research*, 50(2), 2002.

Northrup, A. NT Network Plumbing. IDG Books, pp. 515, 1998.

Rousskov, A. and D. Wessels. Cache Digest. Proceedings of 3rd International WWW Caching Workshop, June 1998.

Tan, Y., V.S. Mookerjee and Y. Ji. Analyzing Document-Duplication Effects on Policies for Browser and Proxy Caching. *INFORMS Journal on Computing*, 18 (4), 2006.

Tawarmalani, M., K. Kannan, P. De, and C. Kumar. Allocating Objects in a Network of Caches: Social Welfare and Incentive Compatibility. *Management Science*, Vol. 55, No. 1, January 2009, pp. 132-147.

Valloppillil, V. and K. W. Ross, "Cache Array Routing Protocol v1.0. Internet Draft", February 1998, http://www.globecom.net/ietf/draft/draft-vinod-carp-v1-03.html.

Wessels, D. and K. Claffy, “ICP and the squid web cache”, IEEE Journal on Selected Areas in Communication, vol. 16, no. 3, pp. 345-357, April 1998. http://ircache.nlanr.net/~wessels/Papers/icp-squid.

Wu, K. and P.S Yu. Local replication for proxy web caches with hash routing. Proceedings of the 8th international Conference on information and Knowledge Management (CIKM), Missouri, November 1999.

Wu, K. and P.S Yu. Load balancing and hot spot relief for hash routing among a collection of proxy caches. Proceedings of 19th IEEE International Conference on Distributed Computing Systems, pp.536-543, 1999.

Zu, M. and J. Subhlok, Home Based Cooperative Web Caching, Seventh Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, FL, July 2003.

Appendix A: Expected Latency

Result A1: In order to derive a simple expression for the expected latency, we will first prove the following result:

$$n!\int_0^{\infty} f(x_1)dx_1 \cdots \int_{x_{n-1}}^{\infty} f(x_n)dx_n \sum_{k=m+1}^{n} \frac{\theta(x_k)}{H_0} = \frac{n-m}{n} \prod_{l=n-m+1}^{n} \left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1} \qquad \text{(A1)}$$

Proof: Plugging equations 1 and 2 to the left hand side of the above expression and simplifying, we get:

$$n!\int_0^{\infty} f(x_1)dx_1 \cdots \int_{x_{n-1}}^{\infty} f(x_n)dx_n \sum_{k=m+1}^{n} \frac{\theta(x_k)}{H_0}$$

$$= n!\left(\frac{1}{\alpha}-1\right)^n \frac{1}{\beta}\frac{1}{H_0} \sum_{k=m+1}^{n} \prod_{j=1}^{n-k} \frac{1}{j(1/\alpha-1)} \prod_{i=n-k+1}^{n} \frac{1}{i(1/\alpha-1)+1} \qquad \text{(A2)}$$

Re-labeling with the new index *k’* = *n* – *k* + 1 (hereafter, the prime is omitted)

$$= n!\left(\frac{1}{\alpha}-1\right)^n \frac{1}{\beta}\frac{1}{H_0} \sum_{k=1}^{n-m} \prod_{j=1}^{k-1} \frac{1}{j(1/\alpha-1)} \prod_{i=k}^{n} \frac{1}{i(1/\alpha-1)+1} \qquad \text{(A3)}$$

By induction, we have,

$$\sum_{k=1}^{l}\prod_{j=1}^{k-1}\frac{1}{j(1/\alpha-1)}\prod_{i=k}^{l}\frac{1}{i(1/\alpha-1)+1}=\frac{\alpha}{(1/\alpha-1)^{l-1}(l-1)!} \tag{A4}$$

Substituting equation A4 into equation A3, we have

$$\begin{aligned}&=n!\left(\frac{1}{\alpha}-1\right)^{n}\frac{1}{\beta}\frac{1}{H_0}\frac{\alpha}{(1/\alpha-1)^{n-m-1}(n-m-1)!}\prod_{l=n-m+1}^{n}\frac{1}{l(1/\alpha-1)+1}\\&=\left(1-\frac{m}{n}\right)\prod_{l=n-m+1}^{n}\left(1+\frac{\alpha}{l(1-\alpha)}\right)^{-1}.\end{aligned} \tag{A5}$$

Hence proved.

Expected Latency: We will now proceed to evaluate the expected latency. The expected latency in cache 2 is formulated in Equation 5. It can be restated as

$$\begin{aligned}EL=&\sum_{j=0}^{\min(L_2,R_1)}p(j)\cdot\sum_{i=0}^{R_1-j}p(i\mid j)\boldsymbol{E}_{\varphi|j,i}\int_{0}^{\infty}f(x_1)dx_1\cdots\int_{x_{n-1}}^{\infty}f(x_n)dx_n\left(\sum_{k=L_2+1}^{R_2+i-1}\frac{\theta(x_k)}{H_0}\varphi_k\cdot D+\right.\\&\left.\sum_{k=R_2+i+1}^{n}\frac{\theta(x_k)}{H_0}(\varphi_k\cdot D+(1-\varphi_k))\right)\end{aligned} \tag{A6}$$

where we have merely substituted $p(j,i)=p(j)\cdot p(i\mid j)$ into Equation 5.

To simplify the above expression, we begin by looking at the probability that there are *j* tagged documents among the first L_2 documents:

$$p(j)=\sum_{i=0}^{R_1-j}p(j,i)=\binom{L_2}{j}\binom{n-L_2}{R_1-j}\cdot\binom{n}{R_1}^{-1}. \tag{A7}$$

Because $\binom{m}{k}$=0 for *k*<0 and *k*>*m*, it follows that *p*(*j*)=0 for $j>\min(L_2,R_1)$. We apply this result later in this section. Using (A7), we can also derive the conditional probability of having *i* tagged documents among the documents ranked from the (L_2+1)*th* to the (R_2 + *i* – 1)*th*

$$p(i\mid j)=\frac{p(j,i)}{p(j)}=\binom{R_2-L_2+i-1}{i}\binom{n-R_2-i}{R_1-j-i}\cdot\binom{n-L_2}{R_1-j}^{-1}. \qquad \text{(A8)}$$

Now note that the $(R_2 + i)^{th}$ document cannot be tagged because it is stored in cache 2. Similarly, given values of *j* and *i*, each document *k* has a probability of

$$\boldsymbol{E}_{\varphi;j,i}\circ\varphi_k=\begin{cases}\dfrac{i}{R_2-L_2+i-1}, & \text{if } L_2+1\le k\le R_2+i-1;\\[2ex] \dfrac{R_1-j-i}{n-R_2-i}, & \text{if } k\ge R_2+i+1.\end{cases} \qquad \text{(A9)}$$

of being tagged. The above expression follows directly from Figure 4. In the Figure, note that there are *i* tagged documents in the no duplication region and $(R_1 - j - i)$ tagged documents among the final $(n - R_2 - i)$ documents. It can be verified that

$$\sum_{i=0}^{R_1-j}p(i\mid j)\frac{i}{R_2-L_2+i-1}=\frac{R_1-j}{n-L_2} \text{ and } \sum_{i=0}^{R_1-j}p(i\mid j)\frac{R_1-j-i}{n-R_2-i}=\frac{R_1-j}{n-L_2}. \qquad \text{(A10)}$$

That is, the probability that any random document outside the duplication-allowed region is tagged (given *j* tagged documents in duplication region) is $(R_1-j)/(n-L_2)$.

By combining Equations A1 and A9, we get

$$T1=\boldsymbol{E}_{\varphi;j,i}\circ n!\int_0^{\infty}f(x_1)dx_1\cdots\int_{x_{n-1}}^{\infty}f(x_n)dx_n\left(\sum_{k=L_2+1}^{R_2+i-1}\frac{\theta(x_k)}{H_0}\varphi_k D+\sum_{k=R_2+i+1}^{n}\frac{\theta(x_k)}{H_0}\left(\varphi_k D+(1-\varphi_k)\right)\right)=$$

$$\frac{i}{R_2-L_2+i-1}\cdot D\left(\frac{n-L_2}{n}\prod_{l=n-L_2+1}^{n}\left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1}-\frac{n-R_2-i+1}{n}\prod_{l=n-R_2-i+2}^{n}\left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1}\right)$$
$$+\left(\frac{R_1-j-i}{n-R_2-i}\cdot D+\frac{n-R_2-R_1+j}{n-R_2-i}\right)\frac{n-R_2-i}{n}\prod_{l=n-R_2-i+1}^{n}\left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1}.$$

(A11)

Averaging the above expression over *i* and simplifying,

$$T2 = \sum_{i=0}^{R_1-j} p(i \mid j)\cdot T1 =$$

$$\frac{R_1-j}{n}\prod_{l=n-L_2+1}^{n}\left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1}\cdot D+\frac{n-R_2-R_1+j}{n}\sum_{i=0}^{R_1-j}p(i \mid j)\prod_{l=n-R_2-i+1}^{n}\left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1} \qquad \text{(A12)}$$

which can be further reduced, after averaging over *j* to yield,[15]

$$EL = \sum_{j=0}^{\min(L_2,R_1)} p(j)\cdot T2 = \sum_{j=0}^{L_2} p(j)\cdot T2 \; =$$

$$\frac{R_1}{n}\frac{n-L_2}{n}\prod_{l=n-L_2+1}^{n}\left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1}\cdot D+\sum_{j=0}^{L_2}\sum_{i=0}^{R_1-j}p(j,i)\frac{n-R_2-R_1+j}{n}\prod_{l=n-R_2-i+1}^{n}\left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1}$$

(A13)

Upon further simplification, we get

$$EL = \frac{R_1}{n}\cdot D\cdot w(n,L_2)+\sum_{j=0}^{L_2}\sum_{i=0}^{R_1-j}p(j,i)\frac{n-R_2-R_1+j}{n-R_2-i}w(n,R_2+i) \qquad \text{(A14)}$$

where $w(x,y)=\left(1-\frac{y}{x}\right)^{\frac{1}{1-\alpha}}$.

When $R << n$, we can expand

$$w(n,R_2+i)\approx 1-\frac{1}{1-\alpha}\frac{R_2+i}{n}+\frac{1}{2}\frac{\alpha}{(1-\alpha)^2}\left(\frac{R_2+i}{n}\right)^2 \qquad \text{(A15)}$$

Substituting this into A14, the expected latency can be approximated by

$$EL = \frac{R_1}{n}\cdot D\cdot w(n,L_2)+\left(1-\frac{R_1}{n}\right)w\left(n,R_2+R_1\cdot\frac{R_2-L_2}{n-R_1}\right) \qquad \text{(A16)}$$

Appendix B

[15] Because *p*(*j*)=0 for $j > \min(L_2,R_1)$, it follows that $\sum_{j=0}^{\min(L_2,R_1)} p(j)\cdot T2 = \sum_{j=0}^{L_2} p(j)\cdot T2$.

Proposition 1: Zero duplication (CARP) is preferable to unmonitored duplication (ICP) for *D*<*D*

vice-versa, where $D_{Th} = \left(\frac{n}{R_1} - 1\right)\left[\frac{\left(1-\frac{R_2}{n}\right)^{\frac{1}{1-\alpha}} - \left(1-\frac{R_2}{n-R_1}\right)^{\frac{1}{1-\alpha}}}{1-\left(1-\frac{R_2}{n}\right)^{\frac{1}{1-\alpha}}}\right]$.

Proof: The expected latency is given by $EL_2 = \frac{R_1}{n} w(n, L_2)\cdot D + \left(1-\frac{R_1}{n}\right) w\left(n, R_2 + R_1 \cdot \frac{R_2 - L_2}{n - R_1}\right)$. In the case of zero duplication, we have

$EL_2(L_2 = 0) = \frac{R_1}{n} D + \left(1-\frac{R_1}{n}\right)\left(1-\cdot\frac{R_2}{n-R_1}\right)^{1/1-\alpha}$. Similarly, with L_2=R_2, we have

$EL_2(L_2 = R_2) = \frac{R_1}{n} D\left(1-\cdot\frac{R_2}{n}\right)^{1/1-\alpha} + \left(1-\frac{R_1}{n}\right)\left(1-\cdot\frac{R_2}{n}\right)^{1/1-\alpha}$. Equating the two expressions for the expected latency, we get the D_{Th} specified in Proposition 1.

Proposition 2: The optimal size of the duplication region is $L_2^* = \max\left(0, n - \frac{n\cdot(n-R_2)}{(n-R_1)\cdot D^{\alpha^{-1}-1} + R_1}\right)$.

Proof: The objective function is

$$\underset{L_2}{Min}\{EL\} = \underset{L_2}{Min}\left\{D\cdot\frac{R_1}{n} w(n, L_2) + \left(1-\frac{R_1}{n}\right) w\left(n, R_2 + R_1\cdot\frac{R_2 - L_2}{n-R_1}\right)\right\} \qquad \text{(B1)}$$

where $w(x, y) = \left(1-\frac{y}{x}\right)^{\frac{1}{1-\alpha}}$. The First Order Condition (FOC) leads to the following expression for the optimal size of the duplication region:

$$L_2^{\ *} = n - \frac{n\cdot(n-R_2)}{\left((n-R_1)\cdot D^{(\alpha^{-1}-1)} + R_1\right)} \qquad \text{(B2)}$$

We will first verify that $L_2^{\ *} \le R_2$ is always satisfied. Plugging in the expression for L_2^* into this condition, we get $L_2^{\ *} \le R_2$ if and only if (iff)

$$n-\frac{n\cdot\left(n-R_2\right)}{(n-R_1)\cdot D^{\frac{1}{\alpha}-1}+R_1}\le R_2 \qquad \text{(B3)}$$

$$\Rightarrow (n-R_1)\cdot D^{\frac{1}{\alpha}-1}+R_1\le n \qquad \text{(B4)}$$

i.e., iff $D^{(\alpha^{-1}-1)}\le 1$. This is guaranteed since $D\le 1$ and $\alpha<1$ by definition. Thus, $L_2^{*}\le R_2$ is always guaranteed. However, the expression may sometimes give negative values. In such cases, the lowest latency for $L_2\in[0,R_2]$ is realized at L_2=0. Thus, it follows that

$$L_2^{*}=\max\left(0,n-\frac{n\cdot(n-R_2)}{(n-R_1)\cdot D^{\alpha^{-1}-1}+R_1}\right). \text{ QED.}$$

Corollary 1: The optimal size of the duplication region is (a) non-decreasing in the inter-cache wait time, *D,* (b) non-decreasing in request locality, (c) non-decreasing in the size of the other cache R_1, and (d) non-decreasing in cache size R_2.

Proof: (a) Impact of Inter-Cache latency *D*:

$$\frac{\partial}{\partial D}\left(n-\frac{n\cdot\left(n-R_2\right)}{\left((n-R_1)\cdot D^{(\alpha^{-1}-1)}+R_1\right)}\right)=\frac{(1-\alpha)\cdot D^{1/\alpha}n(n-R_1)(n-R_2)}{\alpha\left(D^{1/\alpha}(n-R_1)+DR_1\right)^2}\ge 0 \qquad \text{(B5)}$$

Thus, $n-\frac{n\cdot\left(n-R\right)}{\left((n-R)\cdot D^{(\alpha^{-1}-1)}+R\right)}$ is non-decreasing in *D*, implying that L^{*} is also non-decreasing in inter-cache latency *D*.

(b) Impact of request locality (α):

Proof: $$\frac{\partial}{\partial \alpha}\left(n-\frac{n\cdot\left(n-R_2\right)}{\left((n-R_1)\cdot D^{(\alpha^{-1}-1)}+R_1\right)}\right)=\frac{D^{(1+\alpha)/\alpha}n(n-R_1)(n-R_2)\cdot Log(1/D)}{\left(\alpha D^{1/\alpha}(n-R_1)+\alpha DR_1\right)^2}\ge 0 \qquad \text{(B6)}$$

Thus, it follows that L_2^{*} is non-decreasing in α .

(c) Impact of *R₁*, the size of the other cache.

Proof: $$\frac{\partial}{\partial R_1}\left(n - \frac{n\cdot\left(n-R_2\right)}{\left((n-R_1)\cdot D^{(\alpha^{-1}-1)} + R_1\right)}\right) = \frac{D^2 n(n-R_2)(1-D^{\frac{1}{\alpha}-1})}{\left(D^{1/\alpha}(n-R_1)+DR_1\right)^2} \geq 0 \qquad \text{(B7)}$$

Thus, it follows that L_2^* is non-decreasing in R_1.

(d) Impact of cache size R_2.

$$\frac{\partial}{\partial R_2}\left(n - \frac{n\cdot\left(n-R_2\right)}{\left((n-R_1)\cdot D^{(\alpha^{-1}-1)} + R_1\right)}\right) = \frac{Dn}{\left(D^{1/\alpha}(n-R_1)+DR_1\right)} \geq 0 \qquad \text{(B8)}$$

Thus, it follows that L_2^* is non-decreasing in R_2.

Appendix C

The first L_1 documents from cache 1 can be located anywhere in the sorted list at cache 2 (i.e., documents sorted by LRU age at cache 2), while the next (R_1 – L_1) can only be between (L_2 + 1) to n because these documents cannot be in cache 2 by definition. Suppose there are j tagged documents in the first L_2 documents at cache 2 and i tagged documents from (L_2 + 1) to (R_2 + i – 1). Out of these i documents, k are from Cache 1's (R_1 – L_1) region. The total number of ways in which these documents to tag may be chosen is given by:

$$N_1(i,j) = \sum_{k=0}^{i}\binom{L_2}{j}\binom{R_2+i-1-L_2}{i}\binom{i}{k}\binom{n-R_2-i}{R_1-i-j}\binom{R_1-i-j}{R_1-L_1-k} \qquad \text{(C1)}$$

The first term refers to the number of ways in which we can choose the j documents to tag in cache 2's duplication region. The second term selects the i tagged documents between (L_2 + 1) and (R_2 + i – 1). Among these i documents, k are selected and marked as belonging to cache 1's non-duplication region (third term). The fourth term selects and tags the remaining (R_1 – j – i) documents between ($R_2 + i + 1$) and n. Since k documents have already been marked as belonging to the non-duplication region of cache 1, this leaves us with ($R_1 - L_1 - k$) documents yet to be selected for the non-duplication region. These can be selected from the ($R_1 - j - i$) tagged documents in $\binom{R_1 - j - i}{R_1 - L_1 - k}$ ways (fifth term).

Upon simplification,

$$N_1(i,j) = \binom{L_2}{j}\binom{R_2 + i - 1 - L_2}{i}\binom{n - R_2 - i}{R_1 - i - j}\binom{R_1 - j}{L_1 - j} \qquad \text{(C2)}$$

Given j tagged documents in cache 2's duplication region, $i \in [0, R_1 - j]$. Thus, the total number of ways in which we can select the j tagged documents is:

$$N(j) = \sum_{i=0}^{R_1 - j} N(i,j) = \sum_{i=0}^{R_1 - j}\binom{L_2}{j}\binom{R_2 + i - 1 - L_2}{i}\binom{n - R_2 - i}{R_1 - i - j}\binom{R_1 - j}{L_1 - j} \qquad \text{(C3)}$$

$$\Rightarrow N(j) = \binom{L_2}{j}\binom{n - L_2}{R_1 - j}\binom{R_1 - j}{L_1 - j} \qquad \text{(C4)}$$

The value of j can itself be anywhere from 0 to min(L_2, R_1). Thus, the total number of ways in which the documents can be tagged is[16]

$$N = \sum_{j=0}^{L_2}\binom{L_2}{j}\binom{n - L_2}{R_1 - j}\binom{R_1 - j}{L_1 - j} = \binom{n - L_2}{R_1 - L_1}\binom{n + L_1 - R_1}{L_1} \qquad \text{(C5)}$$

Dividing C4 by C5:

$$p(j) = \binom{L_2}{j}\binom{n - L_2}{R_1 - j}\binom{R_1 - j}{L_1 - j} \cdot \binom{n - L_2}{R_1 - L_1}^{-1}\binom{n + L_1 - R_1}{L_1}^{-1} \qquad \text{(C6)}$$

And dividing C2 by C5.

$$p(j,i) = \binom{L_2}{j}\binom{R_2 + i - 1 - L_2}{i}\binom{n - R_2 - i}{R_1 - i - j}\binom{R_1 - j}{L_1 - j} \cdot \binom{n - L_2}{R_1 - L_1}^{-1}\binom{n + L_1 - R_1}{L_1}^{-1} \qquad \text{(C7)}$$

We also have,

[16] Because $\binom{m}{k}$=0 for k<0 and k>m, it follows that $\sum_{j=0}^{\min(L_2,R_1)} N(j) = \sum_{j=0}^{L_2} N(j)$.

$$\sum_{i=0}^{R_1-j} p(i\,|\,j)\frac{i}{R_2-L_2+i-1}=\frac{R_1-j}{n-L_2} \tag{C8}$$

and $$\sum_{i=0}^{R_1-j} p(i\,|\,j)\frac{i}{R_2-L_2+i-1}=\frac{R_1-j}{n-L_2} \tag{C9}$$

The expected latency at cache 2 is given by

$$EL=\sum_{j=0}^{L_2}\left\{p(j)\cdot\sum_{i=0}^{R_1-j}p(i\,|\,j)\cdot T1\right\} \tag{C10}$$

where $T1$ is given by Equation $A11$. Here again, we use $p(j)$=0 for $j>\min(L_2,R_1)$ and set the upper bound for j as L_2. Substituting Equations C6 and C7 into A11 and using results C8 and C9,

$$\begin{aligned}EL=&\frac{1}{n-L_2}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right)\frac{n-L_2}{n}\prod_{l=n-L_2+1}^{n}\left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1}\cdot D\\&+\sum_{j=0}^{L_2}\sum_{i=0}^{R_1-j}p(j,i)\frac{n-R_2-R_1+j}{n}\prod_{l=n-R_2-i+1}^{n}\left(1+\frac{\alpha}{l\cdot(1-\alpha)}\right)^{-1}\end{aligned} \tag{C11}$$

Upon simplification,

$$EL=\frac{1}{n-L_2}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right)\cdot D\cdot w(n,L_2)+\sum_{j=0}^{L_2}\sum_{i=0}^{R_1-j}p(j,i)\frac{n-R_2-R_1+j}{n-R_2-i}w(n,R_2+i) \tag{C12}$$

Equation (C12) can be simplified further when n is sufficiently large as described below.

Approximation: If the ratios R_1/n and R_2/n are small, we can expand

$$w(n,R_2+i)\approx 1-\frac{1}{1-\alpha}\frac{R_2+i}{n}+\frac{1}{2}\frac{\alpha}{(1-\alpha)^2}\left(\frac{R_2+i}{n}\right)^2 \tag{C13}$$

Substituting this into C12 and using the following two results

$$\sum_{j=0}^{L_2}\sum_{i=0}^{R_1-j}p(j,i)\frac{n-R_1-R_2+j}{n-R_2-i}=\frac{n-R_1}{n-L_2}\frac{n+L_1-L_2-R_1}{n+L_1-R_1} \tag{C14}$$

$$\sum_{j=0}^{L_2}\sum_{i=0}^{R_1-j}p(j,i)\frac{n-R_1-R_2+j}{n-R_2-i}\frac{i}{n}=\frac{R_2-L_2}{n(n-L_2)}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right) \tag{C15}$$

we get:

$$\begin{aligned} EL = {} & \frac{1}{n-L_2}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right)\cdot D\cdot w(n,L_2) \\ & +\left(1-\frac{1}{n-L_2}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right)\right)\cdot w\left(n,R_2+\frac{R_1(n+L_1-R_1)-L_1L_2}{n+L_1-L_2-R_1}\frac{R_2-L_2}{n-R_1}\right) \end{aligned} \qquad \text{(C16)}$$

Proposition 3: The best response curves, $L_2^*(L_1)$ and $L_1^*(L_2)$, are non-increasing in L_1 and L_2 respectively. Further, the existence of an equilibrium is guaranteed.

Proof: To prove this result, we apply the Binomial theorem to expand *w*(*x*, *y*):

$$w(x,y)=\left(1-\frac{y}{x}\right)^{1/(1-\alpha)}=1-\frac{1}{(1-\alpha)}\frac{y}{x}+\frac{\alpha}{2(1-\alpha)^2}\left(\frac{y}{x}\right)^2+\ldots \qquad \text{(C17)}$$

For $x >> y$, we can ignore the cubic and other higher order terms to get

$$w(x,y)=\left(1-\frac{y}{x}\right)^{1/(1-\alpha)}=1-\frac{1}{(1-\alpha)}\frac{y}{x}+\frac{\alpha}{2(1-\alpha)^2}\left(\frac{y}{x}\right)^2 \qquad \text{(C18)}$$

When the number of documents, *n*, is much larger than the size of either cache, we can use (C18) to rewrite (C16) as follows:

$$\begin{aligned} EL_2 = {} & \left\{\frac{1}{n-L_2}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right)\left(1-\frac{1}{(1-\alpha)}\frac{L_2}{n}+\frac{\alpha}{2(1-\alpha)^2}\left(\frac{L_2}{n}\right)^2\right)D\right\} \\ & +\left(1-\frac{1}{n-L_2}\left(R_1-\frac{L_1L_2}{n+L_1-R_1}\right)\right)\left(1-\frac{1}{(1-\alpha)}\frac{R_2+\dfrac{(R_1(n+L_1-R_1)-L_1L_2)(R_2-L_2)}{(n+L_1-L_2-R_1)(n-R_1)}}{n}+\right. \\ & \left.\frac{\alpha}{2(1-\alpha)^2}\left(\frac{R_2+\dfrac{(R_1(n+L_1-R_1)-L_1L_2)(R_2-L_2)}{(n+L_1-L_2-R_1)(n-R_1)}}{n}\right)^2\right) \end{aligned} \qquad \text{(C19)}$$

We now apply the conjugate pairs theorem, which states that for the problem $\min_x F(x,a)$, the derivative $\frac{\partial x^*}{\partial a}$ and the cross partial F_{xa} have opposite signs. Taking the cross-partial of (C19) with respect to L_1 and L_2,

$$\frac{\partial^2 EL_2}{\partial L_1 \partial L_2} = \frac{(1-D)\left(n^8(1-\alpha) + n^7\left(3(1-\alpha)L_1 - (5-3\alpha)L_2 - 5(1-\alpha)R_1\right)\right) + O(n^6)}{(1-\alpha)n^2(n-L_2)^2(n+L_1-R_1)^2(n-R_1)(n+L_1-L_2-R_1)^3} \tag{C20}$$

Where $O(n^6)$ denotes terms of order n^6 or lower. For large *n*, these terms can be ignored relative to the higher order terms (i.e., terms of order 7 and 8).

$$\frac{\partial^2 EL_2}{\partial L_1 \partial L_2} = \frac{n^7(1-D)\left(n(1-\alpha) + 3(1-\alpha)L_1 - (5-3\alpha)L_2 - 5(1-\alpha)R_1\right)}{(1-\alpha)n^2(n-L_2)^2(n+L_1-R_1)^2(n-R_1)(n+L_1-L_2-R_1)^3} \tag{C21}$$

As long as $\alpha \leq \left(1 - \frac{2L_2}{n+3L_1-3L_2-5R_1}\right)$, the above expression is greater than or equal to zero for all $L_2 \in [0, R_2]$. Since $n >> \max(R_1, R_2)$, the right hand side of the constraint on α approaches 1. Since α <1, we have $\frac{\partial^2 EL_2}{\partial L_1 \partial L_2} \geq 0$.[17] It follows that L_2^* is weakly decreasing in L_1. In exactly the same manner, we can set up the expression for expected latency at cache 1 and show that L_1^* is weakly decreasing in L_2.

To prove existence of an equilibrium, we compute the second-order derivative of (C16). As before, assuming that $n >> \max(R_1, R_2)$ and ignoring terms with lower exponents of *n*, we get the followed simplified expression:

$$\frac{\partial^2 EL_2}{\partial {L_2}^2} = \frac{2\alpha(1-D)(R_1-L_1)(n-R_1)}{(1-\alpha)(n-L_2)^3(n+L_1-R_1)} \geq 0 \tag{C22}$$

Note that the strategy space is compact and convex for each player ($L_1 \in [0, R_1], L_2 \in [0, R_2]$) and the objective function to be minimized is continuous and quasi convex in each player's own strategy. Thus, it follow's that the game has at least one pure strategy Nash equilibrium (Debreu 1952).

Proposition 4: The best response curves, $L_2^*(L_1)$ and $L_1^*(L_2)$, are both non-decreasing in inter-cache latency, *D*.

[17] For α approaching 1, we can show that the optimal solution is to just use LRU at each cache. This boundary solution ($L_2^* = R_2$) also conforms to the notion of a non-increasing response function.

Proof: To prove the result, we again use the conjugate pairs theorem. Taking the cross-partial of equation (C16),

$$\frac{\partial^2 EL}{\partial D\partial L_2} = -\frac{(n-L_2)^{\alpha/(1-\alpha)}}{n^{1/(1-\alpha)}}\left(\frac{L_1}{n+L_1-R_1}+\frac{\alpha}{1-\alpha}(R_1-\frac{L_1L_2}{n+L_1-R_1})(n-L_2)^{-1}\right)<0 \qquad \text{(C23)}$$

Thus, it follows that $L_2^*(L_1)$ is non-decreasing in *D*. In a similar manner, it can be shown that $L_1^*(L_2)$ is non-decreasing in *D*.

Appendix D

Proposition 5: *With traffic-dependent inter-cache latency, the optimal size of the duplication region is (a) non-decreasing in the inter-cache delay D, (b) non-decreasing in request locality α for τ below some positive threshold, (c) non-decreasing in cache size R$_2$., and (d) non-decreasing in mean service time τ .*

Proof: We apply the conjugate pairs theorem to prove the result.

(a) non-decreasing in the inter-cache delay *D*: $\frac{\partial^2 EL_2}{\partial L_2\partial D} = 1\frac{R_1\left(1-\frac{L_2}{n}\right)^{\alpha/1-\alpha}}{n^2(1-\alpha)} \le 0$. Thus, it follows that

$\frac{\partial L_2^*}{\partial D} \ge 0$.

(b) non-decreasing in request locality α for low τ :

$$\frac{\partial^2 EL_2}{\partial L_2\partial \alpha} = \frac{R_1}{n^2}\frac{1}{(1-\alpha)^2\alpha}\left(1-\frac{L_2}{n}\right)^{\frac{\alpha}{1-\alpha}}\left(-\alpha\frac{R_1}{n}w(n,L_2)\ln\left(w(n,L_2)\right)\psi''+\psi'\ln\left(\psi'\right)\right)$$

Where $\psi' = D + \frac{\tau}{\left(1-\left(\frac{1-\alpha}{\beta}n\tau\right)\frac{R_1}{n}w\right)^2}$ and $\psi'' = \left(\frac{1-\alpha}{\beta}n\tau\right)\frac{2\tau}{\left(1-\left(\frac{1-\alpha}{\beta}n\tau\right)\frac{R_1}{n}w\right)^3}$. Note that when $\tau = 0$, we have $\psi' = D$ and $\psi'' = 0$. Thus, the cross-partial is negative and the optimal L_2^* is non-decreasing in α . For higher values of τ , the sign of the cross partial is determined by the sign of:

$\Gamma_0 = -\alpha\frac{R_1}{n}\left(w\ln w\right)\psi''+\psi'\ln\left(\psi'\right)$. Since $0 < w \le 1$, we have $w\ln w \ge -e^{-1}$. Thus,

$\Gamma_0 \le \Gamma \equiv \alpha \frac{R_1}{n} e^{-1} \cdot \psi'' + \psi' \ln(\psi') = \frac{2\sqrt{\tau}\alpha(1-\alpha)R_1}{e\beta}(\psi' - D)^{3/2} + \psi' \ln(\psi')$. $\Gamma(\psi')$ is a convex function of ψ' in the range $D \le \psi' \le 1$, and $\Gamma(\psi' = D) < 0, \Gamma(\psi' = 1) > 0$. Therefore, there exists a unique threshold for ψ', below which the cross partial is negative and L_2^* is non-decreasing in α. Because ψ' is increasing in τ, it follows that there is a corresponding threshold for τ as well.

(c) non-decreasing in cache size *R*2: $\frac{\partial^2 EL_2}{\partial L_2 \partial R_2} = -\frac{R_1(n-R_1)\left(1-\frac{nR_2 - L_2R_1}{n(n-R_1)}\right)^{1/1-\alpha}\alpha}{\left(n^2 - n(R_2+R_1) + L_2R_1\right)^2(1-\alpha)^2} \le 0$. Thus, it follows that L_2^* is non-decreasing in *R*2.

(d) non-decreasing in cache size *R*1: We can rewrite *EL*2 as

$$EL_2 = \psi\left(\frac{R_1}{n} w(n, L_2)\right) + \left(1 - \frac{R_1}{n}\right) w\left(n, R_2 + R_1 \cdot \frac{R_2 - L_2}{n - R_1}\right)$$

Where the function ψ is convexly increasing, that is, $\psi' > 0$ and $\psi'' > 0$. Taking the cross-partial:

$$\frac{\partial^2 EL_2}{\partial L_2 \partial R_1} = \frac{R_1}{n^2}\frac{1}{1-\alpha}\left(-\left(1-\frac{L_2}{n}\right)^{\frac{\alpha}{1-\alpha}}\frac{1}{n}w(n,L_2)\psi'' \right.$$
$$\left. - \frac{\alpha}{1-\alpha}\left(1-\frac{1}{n}\left(R_2 + R_1 \cdot \frac{R_2 - L_2}{n-R_1}\right)\right)^{\frac{2\alpha-1}{1-\alpha}} \frac{R_2 - L_2}{(n-R_1)^2}\right) < 0$$

Thus, it follows that L_2^* is non-decreasing in *R*1.

(e) non-decreasing in mean service time τ:

$$\frac{\partial^2 EL}{\partial L_2 \partial \tau} = -\frac{\beta^2\left(1-\frac{L_2}{n}\right)^{-1/1-\alpha} R_1\left(\beta\left(1-\frac{L_2}{n}\right)^{-1/1-\alpha} + R_1\tau(1-\alpha)\right)}{n(n-L_2)(1-\alpha)\left(\beta\left(1-\frac{L_2}{n}\right)^{-1/1-\alpha} - R_1\tau(1-\alpha)\right)^3} \le 0$$

. It follows that $\frac{\partial L_2^*}{\partial \tau} \ge 0$.