Web Caching Architectures: Hierarchical and Distributed Caching

Pablo Rodriguez * Christian Spanner Ernst W. Biersack

Institut EURECOM
2229, route des Crêtes, BP 193
06904, Sophia Antipolis Cedex, FRANCE
{rodrigue,spanner,erbi}@eurecom.fr

Abstract

In this paper we analyze the performance of hierarchical and distributed caching architectures. With hierarchical caching, caches are placed at multiple levels of the network. With distributed caching, caches are only placed at the bottom levels of the network and there are no intermediate caches. Our main performance measurement is the expected latency to retrieve a Web document. We find that hierarchical caching has shorter connection times than distributed caching, thus, placing additional copies at intermediate network levels reduces the retrieval latency for small documents. We also find that distributed caching has shorter transmission times than hierarchical caching. Distributed caching has higher bandwidth usage than hierarchical caching. However, the network traffic generated by a distributed scheme is better distributed, using more bandwidth in the lower network levels, which are less congested. We also discuss administrative issues concerning the large scale deployment of distributed caching.

Additionally, we study a hybrid scheme where a certain number of caches cooperate at every level of a caching hierarchy using distributed caching. We find that a "well configured" hybrid scheme can combine the advantages of both hierarchical and distributed caching, reducing the connection time as well as the transmission time. Depending on the hybrid caching architecture, the current parent caches load, and the document size, there is a certain number of caches that should cooperate at each network level to minimize the overall retrieval latency. We propose small modifications of the existing cache-sharing protocols to dynamically determine the degree of cooperation between caches at every level of a hybrid caching scheme.

*P.Rodriguez is supported by the European Commission in form of a TMR (Training and Mobility for Researchers) fellowship. Eurecom's research is partially supported by its industrial partners: Ascom, Cegetel, France Telecom, Hitachi, IBM France, Motorola, Swisscom, Texas Instruments, and Thomson CSF.

1 Introduction

Low latency is crucial for the success of the World Wide Web. One way to reduce the latency to the users and reduce the bandwidth usage is by installing Web caches. The performance of these caches depends on the size of the client community connected to it; the more people using the cache, the higher the probability that a given document has already been requested and is present in the cache. Caches cooperate to increase the probability to hit a document.

One approach to make caches cooperate is by setting up a caching hierarchy. *Hierarchical caching* works as follows. At the bottom level of the hierarchy there are the client caches. When a request is not satisfied by the client cache, the request is redirected to the institutional cache. If the document is not found at the institutional level the request is then forwarded to the regional cache which in turn forwards unsatisfied requests to the national cache. If the document is not found at any cache level, the national cache contacts directly the origin server. When the document is found, either at a cache or at the origin server, it travels down the hierarchy, leaving a copy at each of the intermediate caches. Further requests for the same document travel up the caching hierarchy until the document is hit at any cache level. Hierarchical caching is already a fact of life in much of the Internet [1]. Most ISPs and institutions connected to the Internet have been installing caches to reduce the bandwidth and decrease the latency to their clients [1] [6] [2] [10]. However, there are several problems associated with a caching hierarchy: i) every hierarchy level may introduce additional delays [18] [6], ii) higher level caches may become bottlenecks and have long queuing delays, and iii) multiple document copies are stored at different cache levels.

Recently, a number of researchers have proposed the setup of a totally distributed caching scheme, where there are only caches at the bottom level of the network which cooperate. In *distributed Web*

caching [10] [18], no intermediate caches are set up, and there are only institutional caches which serve each others' misses. In order to decide from which institutional cache to retrieve a miss document, institutional caches keep metadata information about the content of every other cooperating institutional cache. To make the distribution of the metadata information more efficient and scalable, a hierarchical distribution can be used [10] [18]. However, the hierarchy is only used to distribute information about the location of the documents and not to store document copies. With distributed caching most of the traffic flows through low network levels, which are less congested and no additional disk space is required at intermediate network levels. However, a large scale deployment of distributed caching encounters several problems (i.e., high connection times, higher bandwidth usage, administrative issues, etc.). In a smaller scale, where close institutional caches are interconnected through fast links and bandwidth is plenty, distributed caching has very good performance with no additional intermediate cache levels.

In this paper we first develop analytical models to study the performance of a pure hierarchical scheme and compare it with the performance of a pure distributed scheme. Second, we consider a hybrid scheme where caches cooperate at every level of a caching hierarchy using distributed caching and determine the desirable degree of cooperation between caches at a given cache level. We contrast hierarchical and distributed caching according to the latency incurred to retrieve a document. We find that hierarchical caching has lower connection times than distributed caching and, thus, placing redundant copies at intermediate cache levels reduces the connection time. We also find that distributed caching has lower transmission times than hierarchical caching since most of the traffic flows through the less congested low network levels.

Our analysis of the hybrid scheme shows that there is an optimum number of caches that should cooperate at every network level before the request is redirected to the parent cache in the hierarchy or to the origin server. We find that a hybrid scheme with the right number of cooperating caches, can combine the advantages of hierarchical and distributed caching, reducing the connection time as well as the transmission time. The degree of cooperation between caches at every level should be dynamically determined depending on the hybrid caching architecture, the document's size, and the current parent cache and network load. When parent caches or high network levels are congested, distant cooperating caches may offer lower latencies than the parent cache, when the parent caches and high network levels are idle, only few close institutional caches will offer lower retrieval latencies than the parent cache. Thus, we propose small variations of the existing cache-sharing protocols to dynamically determine the degree of cooperation between caches at every level of a hybrid caching scheme.

The rest of the paper is organized as follows. In Section 1.1 we discuss some previous work and different approaches to hierarchical and distributed caching. In Section 2 we describe our specific model for analyzing hierarchical and distributed caching. In Section 3 we provide latency analysis for hierarchical and distributed caching. In Section 4 we present a numerical comparison of both caching architectures. In Section 5 we analyze the hybrid scheme. In Section 6 we propose small variations of the current cache-sharing protocols to dynamically determine the number of cooperating caches. In Section 7 we summarize our findings and conclude the paper.

1.1 Related Work

Hierarchical Web caching cooperation was first proposed in the Harvest project [6]. In the context of distributed caching, the Harvest group also designed the Internet Cache Protocol (ICP) [20], which supports discovery and retrieval of documents from neighboring caches as well as parent caches. Other approach to distributed caching is the Cache Array Routing Protocol (CARP) [19], which divides the URL-space among an array of loosely coupled caches and lets each cache store only the documents whose URL are hashed to it. Povey and Harrison also proposed a distributed Internet cache [10]. In their scheme upper level caches are replaced by directory servers which contain location hints about the documents kept at every cache. A hierarchical metadata-hierarchy is used to make the distribution of these location hints more efficient and scalable. Tewari et al. proposed a similar approach to implement a fully distributed Internet cache where location hints are replicated locally at the institutional caches [18]. In the central directory approach (CRISP) [8], a central mapping service ties together a certain number of caches. In Summary Cache [7], Cache Digest [14], and the Relais project [9] caches inter-exchange messages indicating their content, and keep local directories to facilitate finding documents in other caches.

Concerning a hybrid scheme, ICP allows for cache cooperation at every level of a caching hierarchy. The document is fetched from the parent/neighbor cache with a document copy that has the lowest RTT [20]. Rabinovich et al. [11] proposed to limit the cooperation between neighbor caches to avoid obtaining documents from distant or slower caches, which could

have been retrieved directly from the origin server at a lower cost. In this paper we analyze the advantages and inconvenient of both, distributed and hierarchical caching. We discuss how to better combine hierarchical and distributed caching into a hybrid scheme and study the factors that determine the degree of caches' cooperation to reduce client's perceived latency.

2 The Model

2.1 Network Model

As shown in Figure 1, the Internet connecting the server and the receivers can be modeled as a hierarchy of ISPs, each ISP with its own autonomous administration.

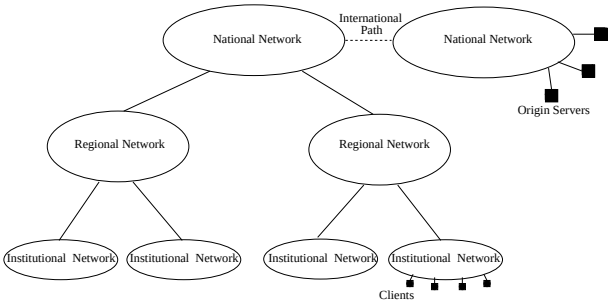


Figure 1: Network topology

We shall make the reasonable assumption that the Internet hierarchy consists of three tiers of ISPs: institutional networks, regional networks, and national backbones. All of the clients are connected to the institutional networks; the institutional networks are connected to the regional networks; the regional networks are connected to the national networks. The national networks are also connected, sometimes by transoceanic links. We shall focus on a model with two national networks, with one of the national networks containing all of the clients and the other national network containing the origin servers.

We model the underlying network topology as a full O-ary tree, as shown in Figure 2. Let O be the nodal outdegree of the tree. Let H be the number of network links between the root node of a national network and the root node of a regional network. H is also the number of links between the root node of a regional network and the root node of an institutional network. Let z be the number of links between a origin server and root node (i.e., the international path). Let l be the level of the tree. $0 \leq l \leq 2H + z$, where $l = 0$ is the institutional caches and $l = 2H + z$ is

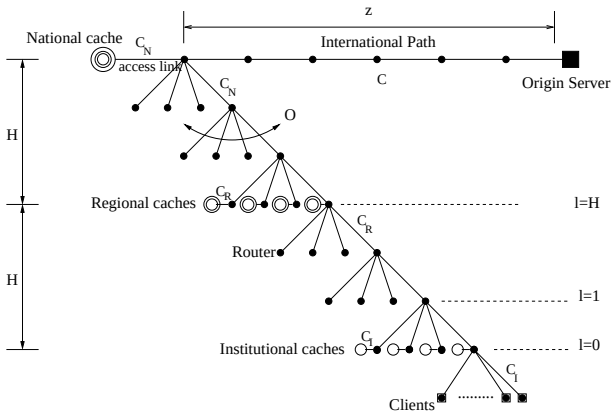


Figure 2: The tree model. Caches placement.

the origin server. We assume that bandwidth is homogeneous within each ISP, i.e. each link within an ISP has the same transmission rate. Let C_I, C_R, and C_N be the *transmission rate* of the links at the institutional, regional, and national networks. Let C be the bottleneck rate on the international path.

2.2 Document Model

We denote the total number of documents in the WWW as N. Denote S the size of a certain document. We assume that all documents change periodically every update period Δ, thus, documents are removed from the caches every Δ seconds. Requests from an institutional cache for document i, $1 \leq i \leq N$, are Poisson distributed with average request rate $\lambda_{I,i}$. Let β_I be the request rate from an institutional cache for all N documents, $\beta_I = \sum_{i=1}^{N} \lambda_{I,i}$. β_I is Zipf distributed [4] [21], that is, if we rank all N documents in order of their popularity, the $i-th$ most popular document has a request rate $\lambda_{I,i}$ given by

$$\lambda_{I,i} = \beta_I \frac{\sigma}{i^{\alpha}}$$

where α takes values between 0.6 and 0.8 [4], and σ is given by

$$\sigma = (\sum_{i=1}^{N} \frac{1}{i^{\alpha}})^{-1}.$$

Assuming that requests for document i are uniformly distributed between all O^{2H} institutional caches, there are $\lambda_{tot} = \lambda_{I,i} \cdot O^{2H}$ total requests for document i.

Note that we assume that the distribution of document requests at every institutional cache is Zipf distributed, however, this does not imply that the distribution of document requests at the regional or national caches is also Zipf distributed. In fact the distribution at the intermediate caches will be filtered

by lower level caches. In our analysis we do not model heterogeneous client communities. If requests from different institutional caches have different request patterns, the hit rate at the intermediate caches will be lower. As we will see in Section 3.2 and Section 4.2, considering homogeneous client communities we obtain analytical hit rates which are very close to those ones reported in real caches.

We consider that each document is requested independently from other documents, so we are neglecting any source of correlation between requests of different documents.

2.3 Hierarchical Caching

Caches are usually placed at the access points between two different networks to reduce the cost of traveling through a new network. As shown in Figure 2, we make this assumption for all of the network levels. In one country there is one national network with one national cache. There are O^H regional networks and every one has one regional cache. There are O^{2H} local networks and every one has one institutional cache.

Caches are placed on height 0 of the tree (level 1 in the cache hierarchy), height H of the tree (level 2 in the cache hierarchy), and height $2H$ of the tree (level 3 in the hierarchy). Caches are connected to their ISPs via *access links*. We assume that the capacity of the access link at every level is equal to the network link capacity at that level, i.e., C_I, C_R, C_N and C for the respective levels. The hit rate for documents at the institutional, regional, and national caches is given by hit_I, hit_R, hit_N.

2.4 Distributed Caching

In the distributed caching scheme, caches are only placed at the institutional level of Figure 2 and no intermediate copies are stored in the network. To share document copies among institutional caches, intermediate network caches are replaced with a metadata hierarchy which contains location hints about about the content of the institutional caches [10]. To avoid lookup latencies at the metadata hierarchy, location hints are replicated locally at every institutional cache [18]. We assume that location hints are instantaneously updated at every institutional cache.

3 Latency Analysis

In this section we model the expected latency to obtain a document in a caching hierarchy and in a distributed caching scheme. We use a similar analysis to the one presented in [12]. The total latency T to fetch a document can be divided into two parts, the *connection time* T_c and the *transmission time* T_t. The connection time T_c is the time since the document is requested by the client and the first data byte is received. The transmission time T_t is the time to transmit the document. Thus, the average total latency is given by

$$E[T] = E[T_c] + E[T_t].$$

3.1 Connection Time

The connection time depends on the number of network links from the client to the cache containing the desired document copy. Let L be the number of links that a request travels to hit a document in the caching hierarchy. We assume that the operating system in the cache gives priority at establishing TCP connections. Let d denote the per-hop propagation delay. The connection time in a caching hierarchy T_c^h is given by

$$E[T_c^h] = 4d \sum_{l \in \{0,H,2H,2H+z\}} P(L = l)(l + 1)$$

where rationale for the $4d$ term is due to the three-way handshake of a TCP connection that increases the number of links traversed before any data packet is sent. To account for the distance between the client and the institutional cache, we consider one more link in the connection time.

Now let L be the network level such that the tree rooted at level L is the smallest tree containing a document copy. The connection time in distributed caching T_c^d is given by

$$\begin{aligned} E[T_c^d] = \quad & 4d \cdot \textstyle\sum_{l=0}^{2H} P(L = l) \cdot (2l + 1) + \\ & 4d \cdot P(L = 2H + z) \cdot (2H + z + 1). \end{aligned}$$

In distributed caching a request first travels up to network level L and then it travels down to the institutional cache with a document copy, thus, accounting for $2L$ links. In hierarchical caching L is the number of links that a request needs to travel to hit the document.

We now proceed to calculate the distribution of L which is the same for hierarchical and distributed caching. To obtain $P(L = l)$ we use $P(L = l) = P(L \geq l) - P(L \geq l + 1)$. Note that $P(L \geq l)$ is the probability that the number of links traversed to meet the document is equal to l or higher. To calculate $P(L \geq l)$ let τ denote the time into the interval $[0, \Delta]$ at which a request occurs. The random variable

τ is uniformly distributed over the interval, thus we have

$$P(L \geq l) = \frac{1}{\Delta} \int_0^{\Delta} P(L \geq l \mid \tau)\ d\tau \qquad (1)$$

where $P(L \geq l \mid \tau)$ is the probability that there is no request for document i during the interval $[0, \tau]$

$$P(L \geq l \mid \tau) = e^{-O^l \cdot \lambda_{I,i} \cdot \tau}. \qquad (2)$$

Combining equation 1 and equation 2 we get

$$P(L \geq l) = \frac{1}{O^l \cdot \lambda_{I,i} \cdot \Delta}(1 - e^{-O^l \cdot \lambda_{I,i} \cdot \Delta}).$$

3.2 Transmission Time

In this section we calculate the transmission time to send a document in a caching hierarchy and in distributed caching. The transmission time of a document depends on the network level L up to which a request travels. Requests that travel through low network levels will experience low transmission times. Requests that travel up to high network levels will experience large transmission times. We make the realistic assumption that the caches operate in a cut-through mode rather than a store-and-forward mode, i.e., when a cache begins to receive a document it immediately transmits the document to the subsequent cache (or client) while the document is being received. We expect capacity misses to be a secondary issue for large-scale cache architectures because it is becoming very popular to have caches with huge effective storage capacities. We therefore assume that each cache has infinite storage capacity.

We now proceed to calculate the transmission time for hierarchical caching $E[T_t^h]$, and the transmission time for distributed caching $E[T_t^d]$. $E[T_t^h]$ and $E[T_t^d]$ are given by:

$$E[T_t^h] = \sum_{l \in \{0,H,2H,2H+z\}} E[T_t^h | L = l] \cdot P(L = l)$$

$$E[T_t^d] = \sum_{l=0}^{2H+z} E[T_t^d | L = l] \cdot P(L = l)$$

where $E[T_t^h | L = l]$ and $E[T_t^d | L = l]$ are the expected transmission times at a certain network level for hierarchical and distributed caching. To calculate $E[T_t^h | L = l]$ and $E[T_t^d | L = l]$ we first determine the aggregate request arrival rate at every network level l for hierarchical caching β_l^h, and distributed caching β_l^d.

For hierarchical caching the aggregate request arrival rate at every network level β_l^h is filtered by the hit rates at the lower caches. Thus, the aggregate request arrival rate generated by hierarchical caching at a link between the levels l and $l+1$ of the network tree is given by

$$\beta_l^h = \begin{cases} \beta_I & l = 0 \\ O^l \beta_I \cdot (1 - hit_I) & 0 < l < H \\ O^l \beta_I \cdot (1 - hit_R) & H \leq l < 2H \\ O^{2H} \beta_I \cdot (1 - hit_N) & 2H \leq l < 2H + z \end{cases}$$

The hit rates at every network level can be calculated using the popularity distribution of the different documents, (i.e., Zipf) and the distribution of L.

$$hit_l = \sum_{i=1}^{N} \frac{\lambda_{I,i}}{\beta_I} \cdot P(L \leq l)$$

For distributed caching, the aggregate request arrival rate at a link between levels l and $l+1$ is filtered by the documents already hit in any institutional cache belonging to the subtree rooted at level l, hit_l. Besides, in distributed caching, the traffic between levels l and $l+1$ needs to be increased by the percentage of requests that were not satisfied in all the other neighbor caches and that are hit in the subtree rooted at level l, $hit_N - hit_l$. Thus, every subtree rooted at level l receives an additional traffic equal to $O^l \beta_I \cdot (hit_N - hit_l)$. Therefore, the request rate between levels l and $l+1$ in distributed caching is given by

$$\beta_l^d = O^l \beta_I \cdot ((1 - hit_l) + (hit_N - hit_l))$$

for $0 \leq l < 2H$ and by $O^{2H} \beta_I \cdot (1 - hit_N)$ for $2H \leq l < 2H + z$.

To calculate the transmission time we model the routers and the caches on the network path from the sending to the receiving host as M/D/1 queues. The arrival rate at a given network level for hierarchical and distributed caching is given by β_l^h and β_l^d. The service rate is given by the link's capacity at every network level (i.e., C_I, C_R, C_N, and C). We assume that the most congested network link from level l to the clients, is the link at level l. Delays on network levels lower than l are neglected. Let $\tilde{S}$ be the average document size of all N documents. The M/D/1 queuing theory gives

$$E[T_t^h | l] = \frac{\tilde{S}}{C_l - \beta_l^h \cdot \tilde{S}} \cdot (1 - \frac{\beta_l^h \cdot \tilde{S}}{2C_l})$$

for the delay at every network level in a caching hierarchy, and

$$E[T_t^d | l] = \frac{\tilde{S}}{C_l - \beta_l^d \cdot \tilde{S}} \cdot (1 - \frac{\beta_l^d \cdot \tilde{S}}{2C_l})$$

for the delay at every network level in a distributed caching scheme.

4 Hierarchical vs Distributed Caching: Numerical Comparison

In this section we pick some typical values for the different parameters in the model to obtain some quantitative results. The following parameters will be fixed for the remainder of the paper, except when stated differently. The network tree is modeled with an out-degree $O = 4$ and a distance between caching levels $H = 3$, yielding $O^H = 64$ regional and $O^{2H} = 4096$ institutional caches. The distance from the top node of the national network to the origin server is set to $z = 10$. Setting a high value for z we model the situation where the cost to access the origin servers is very high.

4.1 Connection Time

The connection time is the first part of the perceived latency when retrieving a document and depends on the network distance to the document. In Figure 3 we show the connection time for distributed and hierarchical caching for different document's popularity. First, we observe that for very unpopular document

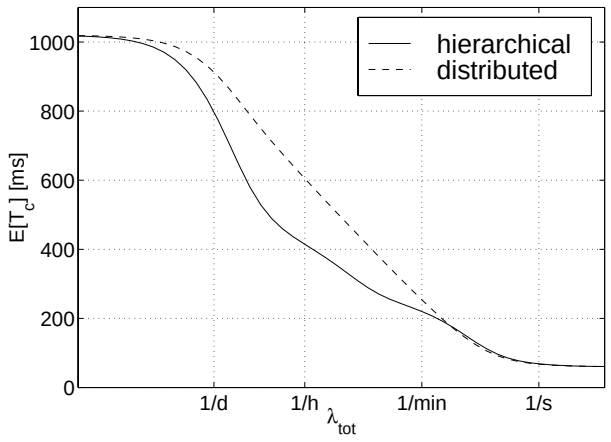


Figure 3: Expected connection Time $E[T_c]$, for hierarchical and distributed caching depending on the document's popularity λ_{tot}. $\Delta = 24$ h, $d = 15$ msec.

(small λ_{tot}) both hierarchical and distributed caching experience high connection times because the request needs to travel to the origin server. As the number of requests for a document increases, the average connection time decreases since there is a higher probability to hit a document at closer caches than the origin server. For all the documents which λ_{tot} ranges between one request per day and one request per minute, a hierarchical caching scheme gives shorter connection times than a distributed caching scheme. Document copies placed at the regional and the national caches in a caching hierarchy reduce the expected network distance to hit a document.

When the document is very popular a distributed caching scheme can benefit from close neighboring copies, reducing the connection time. In a hierarchical caching scheme, however, the document still has to be fetched from the regional cache in case of a miss at the institutional cache. Thus, for very popular documents, the distributed caching scheme has shorter connection times than the hierarchical scheme. However, since the probability that a very popular document is not hit at the local institutional cache is very small, the benefits of distributed caching are almost not appreciable.

4.2 Transmission Time

The second part of the overall latency is the time it takes to transmit the Web document. To calculate the transmission time, we first show the distribution of the traffic generated by distributed caching β_l^d, and hierarchical caching β_l^h at every network level. We consider $N = 250$ million Web documents [3] , which are distributed following a Zipf distribution. We fix the documents update time to $\Delta = 24$h. We consider the total network traffic $O^{2H}\beta_I$ to be equal to 1000 document req/s [5]. We fix the average document size of the N Web documents as $\tilde{S} = 15$ KB [5]. Given these parameters we calculate the hit rates at every cache level, $hit_I = 0.5$, $hit_R = 0.6$, $hit_N = 0.7$, which are very close to those ones reported in many real caches [18] [13].

In Figure 4 we show the traffic generated by hierarchical and distributed caching at every network level. We observe that distributed caching practically doubles the used bandwidth on the lower levels of the network tree, and uses more bandwidth in most part of national network. However, the traffic on the most congested links, around the root node of the national network is reduced to half. Distributed caching uses all possible network shortcuts between institutional caches, generating more traffic in the less congested low network levels.

Once we have presented the traffic generated at every network level we calculate the transmission time for two different scenarios, i) the national network is not congested, and ii) the national network is highly congested. We set the institutional network capacity to $C_I = 100$ Mbps. We consider the same network

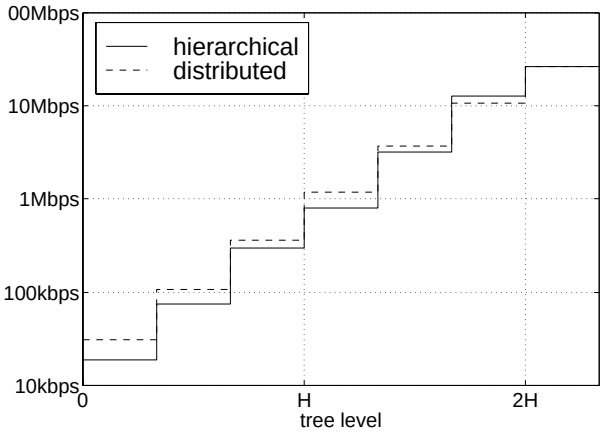
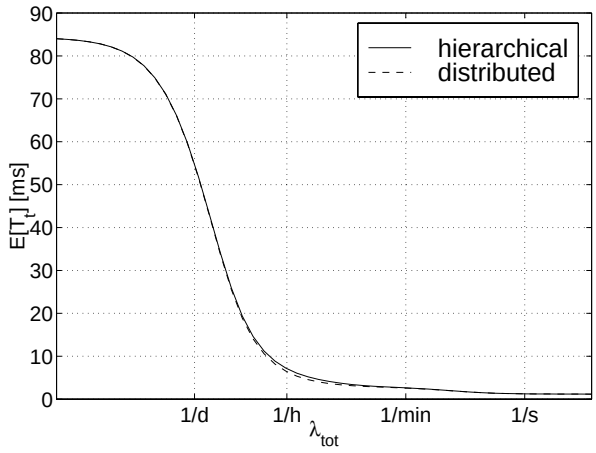
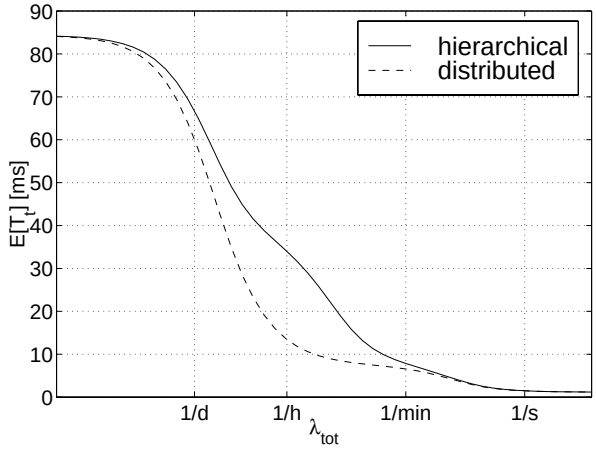


Figure 4: Network traffic generated by distributed and hierarchical caching at every tree level.



(a) Not-congested national network. $\rho = 0.3$.



(b) Congested national network. $\rho = 0.8$.

Figure 5: Expected transmission time $E[T_t]$, for hierarchical and distributed caching. $\tilde{S} = 15$ KB.

link capacities at the regional and national network, $C_N = C_R$. We don't fix the network links capacities at the regional or national networks to certain values, but consider only the degree of congestion ρ under which these links operate (i.e., we vary the utilization of the top national network links in the hierarchical caching scheme $\rho = \frac{\beta^h_{2H}\tilde{S}}{C_N}$). The international path is always very congested and has a utilization of $\frac{\beta_I O^{2H}(1-hit_N)\tilde{S}}{C} = 0.95$.

In Figure 5(a) we show the transmission time for the case where the national network is not congested ($\rho = 0.3$). The only bottleneck on the path from the client to the origin server is the international path. We observe that the performance of both hierarchical and distributed caching is very similar because there are no highly congested links.

In Figure 5(b) we contrast the performance of hierarchical and distributed caching in a more realistic scenario where the national network is congested, $\rho = 0.8$. We observe that both, hierarchical and distributed caching have higher transmission times than in the case when the national network is not congested (Figure 5(a)). However, the increase in the transmission time is much higher for hierarchical caching than for distributed caching. Distributed caching gives shorter transmission times than hierarchical caching because many requests travel through lower network levels. Similar results are also obtained in the case that the access link of the national cache is very congested. In this situation the transmission time in distributed caching remains unchanged, while the transmission time in hierarchical caching increases considerably [16].

4.3 Total Latency

The total latency is the sum of the connection time and the transmission time. For large document sizes, the transmission time is more relevant than the connection time. For small document sizes, the transmission time is very small and the connection time has a higher relevance. Next, we present the total latency, for different document sizes in both hierarchical and distributed caching. We will study the total latency in the scenario where the top nodes of the national network are highly congested. In Figure 6 we ob-

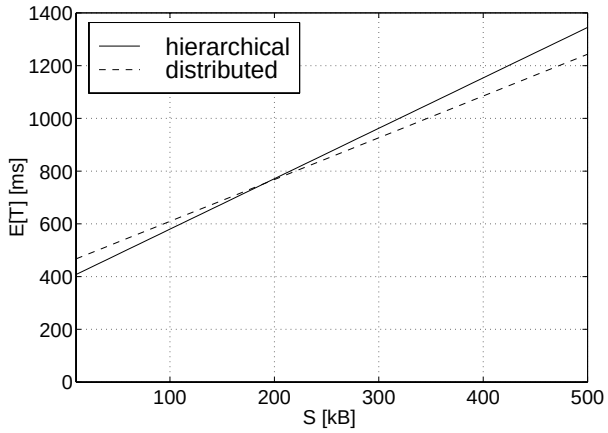


Figure 6: Average total latency depending on the document size S. National network is congested, $\rho = 0.8$

serve that hierarchical caching gives lower latencies for documents smaller than 200 KB because hierarchical caching has lower connection times than distributed caching. However, distributed caching gives lower latencies for higher documents because distributed caching has lower transmission times than hierarchical caching. The size-threshold depends on the degree of congestion in the national network. The higher the congestion, the lower is the size-threshold from which distributed caching has lower latencies than hierarchical caching.

As we have seen distributed caching can decrease the retrieval latency of large documents and reduce the bandwidth usage at high network levels. However, full deployment of distributed caching encounters several problems. Given that in a distributed caching scheme documents are retrieved from neighbor institutional caches, the experienced latency depends not only on the bandwidth of the requesting institutional cache, but also on the bandwidth of the neighbor cache that is contacted. Thus, in a distributed caching scheme investing into a higher capacity Internet connection will not result in any benefit when documents are hit in neighbor caches with smaller connection capacities.

In order to increase the local hit rates local ISPs could increase the disk space of their cache and thus store more documents. However, in distributed caching the more documents a given institutional cache stores, the higher it is the number of external requests it will receive from other neighbor institutional caches. Thus, investing in larger disks to save bandwidth and reduce latency can eventually result in *more* incoming traffic and *longer* queuing delays at the local cache. Additionally, distributed caching has longer connection times and higher bandwidth usage in the low network levels. Nevertheless, distributed caching can be used in a smaller scale where caches are interconnected with short distances and plentiful bandwidth, i.e. among the caches on a campus or in a metropolitan area.

5 A Hybrid Caching Scheme

In this section we consider a hybrid caching scheme where a certain number of caches cooperate at every network level of a caching hierarchy. We study the impact of k, the number of caches that cooperate at every network level, in the client's perceived latency. The analysis of the hybrid scheme that can be found in [16].

5.1 Connection Time

Next, we present the connection time in a hybrid scheme depending on the number of cooperating caches k at every level. The number of cooperating caches at every cache level can range from 1 (no cooperation) to $O^H = 64$ (all neighbor caches in the same cache level cooperate). In Figure 7, we show the average connection time for all N Web documents, depending on the number k of cooperating caches. We

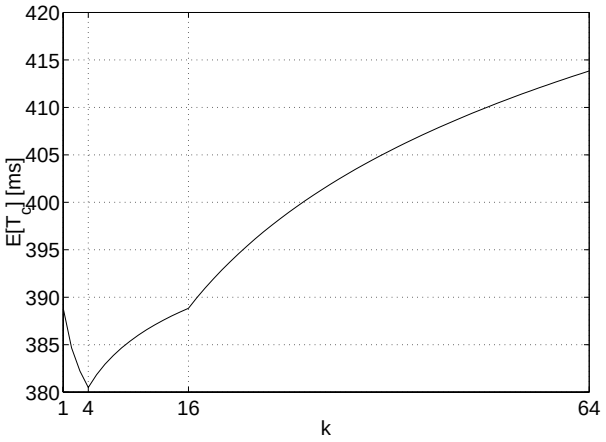


Figure 7: Connection time depending on the number of cooperating caches at every cache level in a hybrid scheme.

observe that when the number of cooperating caches is very small, the connection time is high. The probability that a document is found in few close neighbor caches is very small, thus, most of the requests are satisfied by the parent cache at a distance of H hops. When the number of cooperating caches increases, the connection time decreases up to a minimum. This is

due to the fact that the probability to hit a document at neighbor caches which are closer than the parent cache increases. However, when the number of cooperating caches increases over a certain threshold $k_c = 4$, the connection time increases very fast because documents are requested from cooperating caches at distant network levels. There is, therefore, an optimum number of caches that should cooperate at every cache level to minimize the connection time. The optimum number of cooperating caches k_c that minimize the connection time is given number of caches that are at a close network distances than the parent cache, $k_c = O^{\lfloor H/2 \rfloor}$.

In Figure 8, we present the connection time for i) a hybrid scheme with the optimum number of cooperating caches k_c, ii) distributed caching, and iii) hierarchical caching. We observe that a hybrid scheme with k_c cooperating caches has lower connection times than distributed caching and even lower connection times than hierarchical caching for a large range of documents' popularities.

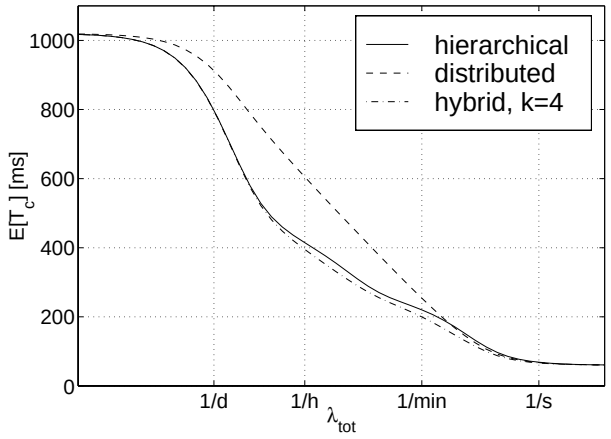


Figure 8: Connection time in a hybrid caching scheme with the optimal number of cooperating caches k_c.

5.2 Transmission Time

Next, we analyze the transmission time in a hybrid scheme and calculate the optimum number k_t of cooperating caches at every network level that minimize the transmission time.

In Figure 9 we plot the transmission time in a hybrid scheme for all N Web documents depending on the number of cooperating caches at every cache level. We consider the case where the top links of the national network are not congested ($\rho = 0.3$), that is, the only bottleneck in the path from the origin server to the client is the international path, and the case where the top links of the national network are congested ($\rho = 0.8$). Similar results are also obtained for the case that the access link of the national cache is congested [16].

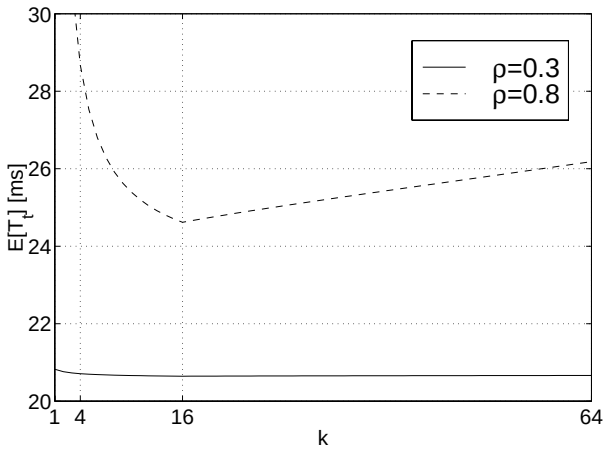


Figure 9: Average transmission time depending on the number of cooperating caches in a hybrid scheme. National network is not congested, $\rho = 0.3$. National network is congested, $\rho = 0.8$. $\tilde{S} = 15$ KB.

We observe that for the case where the national network is not congested varying the number of cooperating caches k at every cache level hardly influences the transmission time. However, when the national network is congested, the transmission time strongly depends on the number of cooperating caches at every cache level. If the number of cooperating caches is very small, there is a low probability that the document can be retrieved from close neighbor caches. The document is retrieved most of the times from the parent cache through the highly congested top-level links. As the number of cooperating caches increases, the probability to hit the document at close neighbor caches connected by fast links increases, and thus, the transmission times are lower. If the number of cooperating caches increases over a certain threshold $k_t = 16$, the transmission time increases again because documents are hit in distant neighbor caches through highly congested top-level links. The optimum number of cooperating caches k_t that minimizes the experienced transmission time, depends on the number of cooperating caches that can be reachable avoiding the congested links. In the case where the top-level links of the national network are congested, the optimum number of cooperating caches at every cache level is $k_t = 16$. This value corresponds to the number of regional caches that can cooperate without traversing the national top-level links, $k_t = O^{H-1}$.

In Figure 10, we present the transmission time $E[T_t]$ for i) a hybrid scheme with the optimum number of cooperating caches k_t, ii) distributed caching, and iii) hierarchical caching. We observe that a hybrid scheme with k_t cooperating caches has lower transmission times than hierarchical caching. We also observe that a hybrid scheme has even lower transmission times than distributed caching since it reduces more the traffic around the high network levels [16].

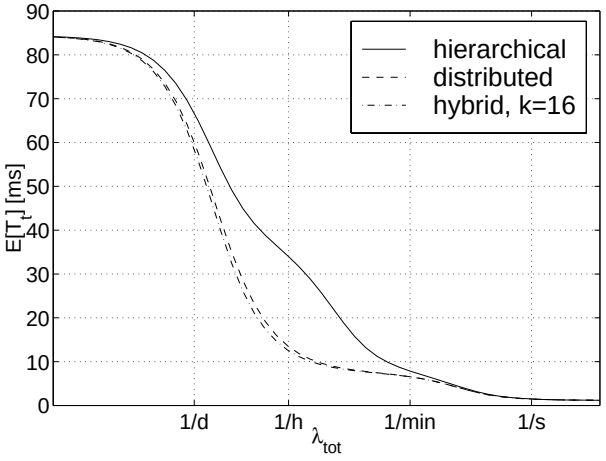


Figure 10: Transmission time for a hybrid caching scheme with the optimal number of cooperating caches k_t. National network is congested, $\rho = 0.8$. $\tilde{S} = 15$ KB.

Thus, dynamically choosing the number of cooperating caches, a hybrid scheme can have as low connection times as hierarchical caching, and as low transmission times as distributed caching.

5.3 Total Latency

Depending on the document size there is an optimum number of caches that should cooperate at every cache level to minimize the total latency. For small documents the optimum number of cooperating caches is close to k_c, since choosing k_c cooperating caches minimizes the connection time. For large documents the optimum number of cooperating caches is close to k_t, since choosing k_t cooperating caches minimizes the transmission time. For any document size, the optimum number of cooperating caches k_{opt} that minimizes the total retrieval latency is such that $k_c \leq k_{opt} \leq k_t$.

In Figure 11 we plot the optimum number of caches k_{opt} that should cooperate at every network level to minimize the total retrieval latency depending on the document size. We choose the case where the top-level links of the national network are highly congested, thus the optimum number of caches that minimizes the transmission time is $k_t = 16$. In Figure 11 we observe that k_{opt} ranges between $k_c = 4$ and $k_t = 16$. For documents smaller than several KBytes, only $k_c = 4$ caches should cooperate at every cache level. For documents larger than several cents of KBytes, $k_t = 16$ caches should cooperate at every cache level.

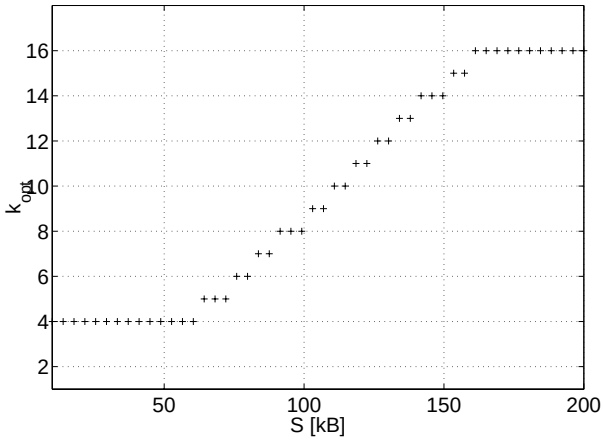


Figure 11: Optimum number of cooperating caches k_{opt}, depending on the document size S. National network is congested, $\rho = 0.8$.

In Figure 12 we show the total retrieval latency for a large document ($S = 200$ KB) and the optimal number of cooperating caches $k_{opt} = k_t = 16$. We see that a hybrid scheme with the optimal number of cooperating caches at every cache level has lower overall retrieval latencies than distributed and hierarchical caching for a large range of document's popularities.

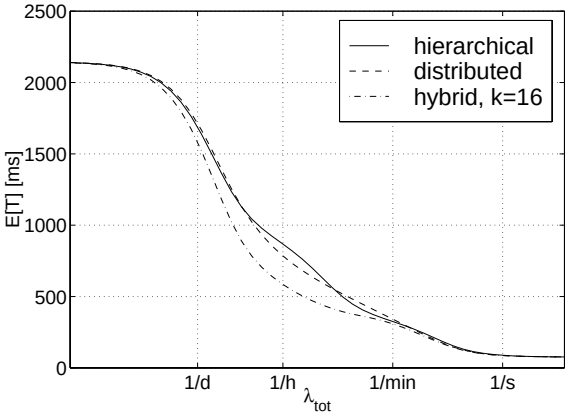


Figure 12: Total latency to retrieve a large document in a hybrid caching scheme with the optimum number of cooperating caches $k_{opt} = k_t = 16$. National network is congested, $\rho = 0.8$. $S = 200$ KB.

6 Suggestions to improve cache-sharing protocols

In this section we propose small modifications to the existing cache-sharing protocols [7] [14] [9] that take into account the results presented in this paper to dynamically select the best cache/server to fetch the document.

In the case that parent caches and top-level links of the network are *not* congested, the transmission time from a neighbor cache is very similar to the transmission time from a parent cache. In this situation the number of cooperating caches should be configured to minimize the connection time, that is, take only those neighbor caches that have an smaller network distance (i.e. round-trip-time) than the parent caches. In the more probable situation where parent caches or top-level links are highly congested, the optimal number of cooperating caches depends on the degree of congestion on the caches/network and on the document size. Using only the round trip time to select among caches does not reflect the achievable transmission rate between two caches, specially the round-trip-time does not consider the application-level load of the caches [15]. A cache can have a very small round-trip-time and at the same time a very small transmission rate, that is, the cache is located at a close network distance but it is highly congested [17]. Using the RTT as the only selection factor distant neighbor caches are never selected, even if they could give shorter transmission times. We suggest that existing cache-sharing protocols [14][7] use also the available transmission rate μ between two caches as a hint to decide from which cache to retrieve a document.

All the analysis presented in this paper for parent caches in a caching hierarchy also applies to origin servers. Caches should keep cache digests including the document size, the round-trip time and the transmission rate of neighbor caches, parent caches, and origin servers. When a request comes for a document of size S the cache could calculate the expected retrieval latency from the neighbor caches with a document copy, from the parents caches, and from the origin servers. The total retrieval latency could be calculated as $2RTT + \frac{S}{\mu}$, where $2RTT$ accounts for the connection time of a TCP connection, and $\frac{S}{\mu}$ for the transmission time of the document. The cache should request the document from the cache/server with the lowest total retrieval latency. We have experimental results showing that if caches also take into account the transmission time to select the cache/server where to fetch the document, client's perceived latency can be reduced by a factor of 1.6 [16].

7 Conclusion

In this paper we have analyzed the performance of two different caching architectures, hierarchical and distributed caching. We have also analyzed a hybrid scheme where caches cooperate at every network level of a caching hierarchy. Hierarchical Web caching achieves shorter connection times by placing document copies at intermediate network levels close to many receivers. A caching hierarchy also reduces the bandwidth usage. However, in a caching hierarchy higher level caches can easily become highly congested. On the other hand, distributed Web caching achieves shorter transmission times since it distributes the network traffic away from the congested links. Distributed caching has very good performance in well interconnected areas without requiring any intermediate cache levels. However, the deployment of distributed caching in a large scale encounters several problems (i.e., large connection times, high bandwidth usage, administrative issues).

A hybrid caching scheme can combine the advantages of both hierarchical and distributed caching, reducing the connection time as well as the transmission time. The degree of cooperation between caches at the same level is configurable, and can be tuned to minimize the overall retrieval latency as well as the bandwidth usage depending on the congestion of the network, the parent cache/server load, and the document's size.

References

[1] M. Baentsch, L. Baum, G. Molter, S. Rothkugel, and P. Sturm, "World Wide Web Caching: The Application-Level View of the Internet", *IEEE Communications Magazine*, pp. 170–178, June 1997.

[2] A. Bestavros et al., "Application-level Document Caching in the Internet", In *Proc. of SDNE'95: The second International Workshop on Services in Distributed and Network Environments*, Whistler, Canada, June 1995.

[3] K. Bharat and A. Broder, "A Technique for Measuring the Relative Size and Overlap of Public Web Search Engines", In *Seventh International WWW Conference*, Brisbane Australia, April 1997.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "On the Implications of Zipf's Law for Web Caching", In *Proceedings of IEEE INFOCOM'99*, New York, USA, March 1999.

[5] E. A. Brewer, P. Gauthier, and D. McEvoy, "The Long-Term Viability of Large-Scale Caching", In *3rd International WWW Caching Workshop*, Manchester, UK, June 1998.

[6] A. Chankhunthod et al., "A Hierarchical Internet Object Cache", In *Proc. 1996 USENIX Technical Conference*, San Diego, CA, January 1996.

[7] L. Fan, P. Cao, J. Almeida, and A. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol", pp. 254–265, Feb 1998, SIGCOMM'98.

[8] S. Gadde, M. Rabinovich, and J. Chase, "Reduce, Reuse, Recycle: An Approach to Building Large Internet Caches", In *The Sixth Workshop on Hot Topics in Operating Systems (HotOS-VI)*, May 1997.

[9] M. Makpangou and Éric Bérenguier, "Relais : un protocole de maintien de cohérence de caches Web coopérants", In *Proceedings of the NoTeRe colloquium*, March 1997.

[10] D. Povey and J. Harrison, "A Distributed Internet Cache", In *Proceedings of the 20th Australian Computer Science Conference*, Sydney, Australia, February 1997.

[11] M. Rabinovich, J. Chase, and S. Gadde, "Not all hits are created equal: cooperative proxy caching over a wide-area network", In *3rd International WWW Caching Workshop*, Manchester, UK, June 1998.

[12] P. Rodriguez, K. W. Ross, and E. W. Biersack, "Distributing Frequently-Changing Documents in the Web: Multicasting or Hierarchical Caching", *Computer Networks and ISDN Systems. Selected Papers of the 3rd International Caching Workshop*, pp. 2223–2245, 1998.

[13] A. Rousskov, "On Performance of Caching Proxies", In *ACM SIGMETRICS*, Madison, USA, September 1998.

[14] A. Rousskov and D. Wessels, "Cache Digest", In *3rd International WWW Caching Workshop*, June 1998.

[15] M. Sayal, Y. Breibart, P. Scheuermann, and R. Vingralek, "Selection Algorithm for Replicated Web Servers", In *Workshop on Internet Server Performance, SIGMETRICS*, Madison, USA, June 1998.

[16] C. Spanner, "Evaluation of Web Caching Strategies: Distributed vs. Hierarchical Caching", M.S. Thesis, University of Munich/Institut Eurécom, Sophia Antipolis, France, November 1998.

[17] O. Stoeckle and R. Pletka, "Web caching: How to select your best siblings", Project report, EURECOM, 1998.

[18] R. Tewari, M. Dahlin, H. M. Vin, and J. S. Kay, "Beyond Hierarchies: Design Considerations for Distributed Caching on the Internet", In *Proceedings of the ICDCS '99 conference*, Austin, Texas, May 1999.

[19] V. Valloppillil and K. W. Ross, "Cache Array Routing Protocol v1.1. Internet Draft", February 1998, http://ds1.internic.net/internet-drafts/draft-vinod-carp-v1-03.txt.

[20] D. Wessels and K. Claffy, "Application of Internet Cache Protocol (ICP), version 2", Internet Draft:draft-wessels-icp-v2-appl-00. Work in Progress., Internet Engineering Task Force, May 1997.

[21] G. K. Zipf, *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*, Addison-Wesley, Reading, MA, 1949.