Bringing the Web to the Network Edge: Large Caches and Satellite Distribution

Pablo Rodriguez Ernst W. Biersack
Institut EURECOM
2229, route des Crêtes. BP 193
06904, Sophia Antipolis Cedex, FRANCE
{rodrigue, erbi}@eurecom.fr
Tel: (33) 04 93 00 2673
Fax: (33) 04 93 00 2627

Abstract

In this paper we discuss the performance of a document distribution model that interconnects Web caches through a satellite channel. During recent years Web caching has emerged as an important way to reduce client-perceived latency and network resource requirements in the Internet. Also a satellite distribution is being rapidly deployed to offer Internet services while avoiding highly congested terrestrial links. When Web caches are interconnected through a satellite distribution, caches end up containing all documents requested by a huge community of clients. Having a large community of clients connected to a cache, the probability that a client is the first one to request a document is very small, and the number of requests that are hit in the cache increases. In this paper we develop analytical models to study the performance of a cache-satellite distribution. We derive simple expressions for the hit rate of the caches, the bandwidth in the satellite channel, the latency experienced by the clients, and the required capacity of the caches. Additionally, we use trace driven simulations to validate our model and evaluate the performance of a real cache-satellite distribution.

1 Introduction

The growth of the World Wide Web overloads popular servers, increases the network traffic, and causes slow responses to the clients. To alleviate these problems, Web caching is being extensively deployed in the Internet. With Web caching when a client requests a document for the first time, the document is delivered directly from the origin server and a copy of the document is stored in the cache. Further requests for the same document are satisfied directly from the cache. Web caching happens at different network levels: Clients have local caches that satisfy multiple requests for the same document coming from the same client (i.e. temporal locality). Local ISPs have institutional caches that satisfy requests for the same document coming from different clients (i.e., geographical locality). Requests satisfied by a cache are called *hits*. Requests not satisfied by a cache are called *misses*. misses can be classified into:

- *First-Access*: misses occurring when requesting documents for the first time.
- *Capacity*: misses occurring when accessing documents previously requested but discarded from the cache due to space limitations.
- *Updates*: misses occurring when accessing documents previously requested that have expired in the meantime.

- *Non-cacheable*: misses occurring when accessing documents that need to be delivered from the origin server (e.g. dynamic documents generated from cgi-bin scripts or fast changing documents)

Even when a cache has infinite storage capacity, the number of misses in an institutional cache can be very high (50%-70%) [3] [4]. Recent studies have shown that non-cacheable documents typically account for 10%-20% of all requests[28][22]. Furthermore, update misses account for approximately 9% of all requests [28]. Thus, there is a large percentage of requests that result in first-access misses (30%-50%), in particular when a small client population is connected to the cache.

One way to reduce the number of both first-access, and update misses is to preload the cache with new documents and document updates, expecting that clients are likely to request them later. Prepopulating caches, however, requires additional disk space and may waste network bandwidth. Disk space may not be such a problem since disk capacity is increasing at rate of 60% per year and large disks are becoming increasingly cheaper [7][16]. A more serious problem arises when network bandwidth is wasted by prefetching documents that no client requests through highly congested and expensive Internet links. An alternative way to preload Web caches is to use a satellite distribution, which has fewer losses and congestion problems than a distribution in the terrestrial Internet. Also, a satellite distribution can reach a very large number of receivers with low cost and relatively little effort- adding a new additional receiver does not increase the cost of transmission.

A *cache-satellite* distribution works as follows. Clients are connected to caches that have a satellite receiving dish. When there is a miss at any cache, the cache obtains the document from the origin server and reports the missed URL to a *master* site that is equipped with a satellite transmitting antenna. The master site fetches itself the document from the origin server and transmits the document over the satellite channel. As a result, *all* caches connected to the cache-satellite distribution receive the broadcasted document and can use it to satisfy local requests. The probability that a cache is the first cache asking for a document becomes very small, thus, reducing significantly the number of first-access and update misses. As more caches join the satellite distribution, the aggregation of clients is higher and the number of misses in one cache is smaller. After a certain period of time, caches with huge storage capacity can end up containing most of the documents in the Web. Recently, several companies (e.g., Sky Cache [27]) have started to offer such a cache-satellite distribution service. For a more detailed description of the technical details of a cache-satellite distribution see [27][24].

In this paper we investigate the feasibility and performance of a cache-satellite scheme. In the first part of the paper we develop analytical models to calculate upper bounds on the performance of a cache-satellite distribution. Using an analytical model, we can study the effect of many different parameters over a large range of values and examine the implications of future trends in Web traffic. In particular, we derive expressions to calculate the maximum achievable hit rate in a cache connected to a cache-satellite satellite distribution and study how the hit rate varies with the client population. We also analyze the latency experienced by clients when caches cooperate through a satellite distribution and when caches cooperate through a caching hierarchy, which is commonly used in the Internet to make caches cooperate [9]. In addition, we calculate the storage capacity needed to store all documents pushed through the satellite as well as the bandwidth requirements for the satellite link. Exploiting the highly skewed distribution of Web documents, where very few documents account for most of the requests, we also study the performance of a satellite distribution where only those documents that see a certain number of requests during the lifetime of the document are pushed through the satellite link. We study the impact of this simple filtering policy on the hit rate, bandwidth, and disk space.

In the second part of the paper we use trace driven simulations to evaluate a real cache-satellite distribution. In particular we examine logs from an ISP in the USA (AYE) that is connected to the SkyCache satellite distribution [27]. This cache-satellite distribution covers the USA and Europe and comprises a large number of ISPs. We study the hit rate improvement offered by a cache-satellite distribution and the additional disk space requirements. We then use NLANR [3] traces to explore the potential of a cache-satellite distribution with a higher number of clients. For this purpose, we analyze traces from AYE and NLANR that were collected during the same period of time and simulate a scenario where NLANR caches also get connected to the cache-satellite distribution.

Our results show that a cache-satellite distribution has important benefits for ISPs with a small client population. ISPs with few clients can achieve very high hit rates *locally*, without having to install and maintain any cache-cooperation protocol for creating a larger receiver population. A cache-satellite distribution, automatically federates a large number of ISP caches with minimum configuration and very little effort, increasing the hit ratios and providing fast access to a large number of documents. In the case of an ISP with a large client population, connecting itself to a cache-satellite distribution does not increase the hit rate significantly. However, a cache-satellite distribution can significantly reduce the cost of any ISP to fill the cache with many documents, since documents reach the cache through the satellite link, allowing the more expensive terrestrial links to be used for other services.

The rest of the paper is organized as follows. Section 2 discusses related work and similar approaches to increase the performance of Web caches. In Section 3 we present the model of a cache-satellite distribution that will be used for our analysis. Section 4 derives expressions to evaluate the performance of a cache-satellite distribution. In Section 5 we explore the impact of several model parameters on latency, hit rate, disk capacity, and bandwidth. Section 6 uses trace driven simulation to study the performance of a cache-satellite distribution. The results obtained in Section 6 are also used to validate the model developed in Section 3. The paper ends with a conclusion and summary of the results.

2 Related Work

Web caching has been recognized as one of the most important techniques to help scaling the Internet. The hit rate of a Web cache is a function of the client population it manages. During the last years there has been extensive research on how to make caches cooperate to increase the total effective client population, increase the hit ratios, and reduce the document-access latency. Web caching cooperation was first proposed in the context of the Harvest project [10], that designed the Internet Cache Protocol (ICP) [31], which supports discovery and retrieval of documents from neighboring caches. Today, many caches have established hierarchies of caches that cooperate via ICP [3]. Other approaches to make caches cooperate have been proposed recently, such as the Cache Array Routing Protocol (CARP) [29], the central directory approach (CRISP) [14], Summary Cache [12], Cache Digest [26], the Relais project [21]. All these approaches use different strategies to share meta-information indicating the location of Web documents in Web caches.

Cache cooperation needs a careful selection of the cooperating caches and in some cases a whole new application-layer routing infrastructure with intermediate caches in the path from the local ISP to the origin server [30] [32]. Additionally, requests not hit in the local ISP need to travel to other ISP caches, which may be overloaded or connected through congested links. If many documents could be prefetched into local caches, most of the requests could be satisfied locally and no intermediate caches or cooperating protocols would be needed. There have been several approaches to preload documents into Web caches [13] [20] [18]. However, all these approaches use the terrestrial Internet links and incur a high cost if the prefetched documents are not requested. In this paper, on the other hand, we present a scheme where documents requested by a large community are pushed into local caches using a satellite distribution, which is cost-efficient and does not require any inter-cache cooperation.

3 The Model

3.1 Internet Topology

As shown in Figure 1, the Internet connecting the server and the clients can be modeled as a hierarchy of ISPs, each ISP with its own autonomous administration. We shall make the reasonable assumption that the Internet hierarchy consists of three levels of ISPs: institutional, regional, and national. All of the clients are connected to the institutional ISP; the institutional ISPs are connected to the regional networks; the regional networks are connected to the national ISP.

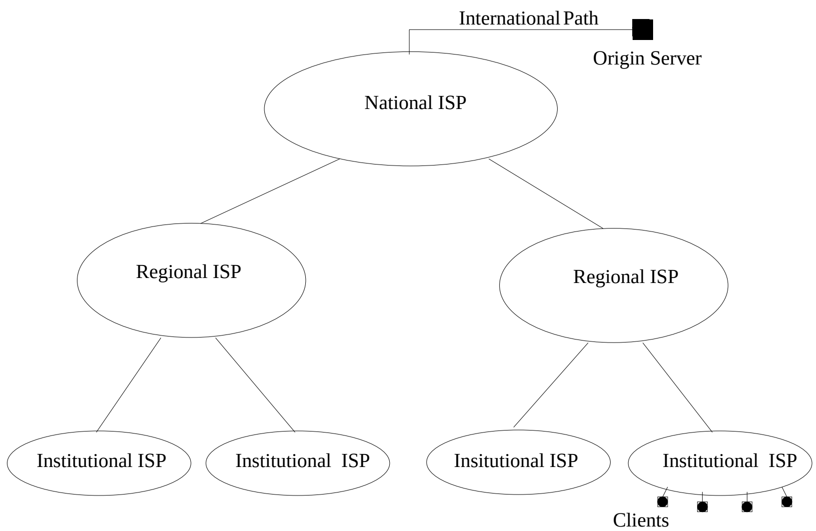


Figure 1: Internet topology.

We model the underlying network topology in Figure 2 as a full O-ary tree. Let O be the nodal outdegree of the tree. Let H be the number of network links between the root node of a national ISP and the root node of a regional ISP. We also suppose that H is the number of links between the root node of a regional ISP and the root node of a institutional ISP. Let z be the number of links between a origin server and the root node of the tree (i.e., the international path).

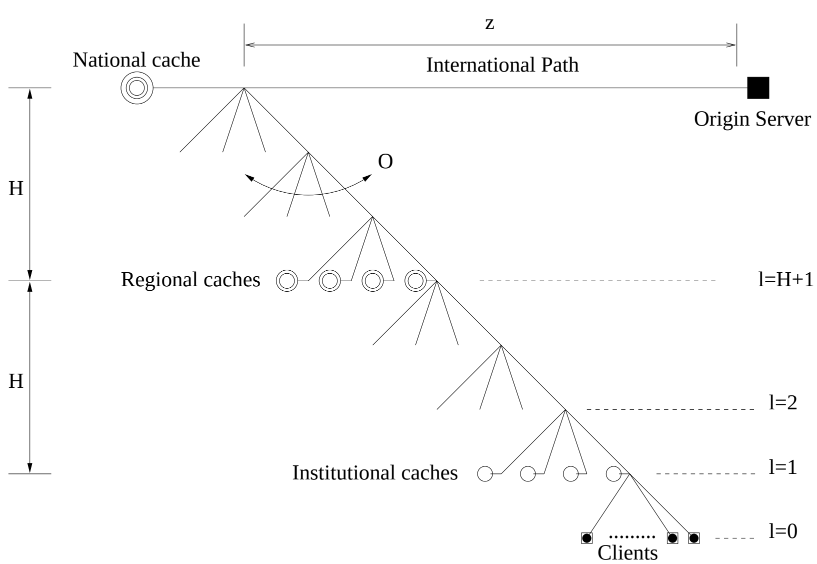


Figure 2: The tree model.

3.2 Hierarchical Caching in the Internet

To compare the performance of a cache-satellite distribution to a cache-cooperating scheme, we consider a hierarchical caching cooperation, which is one popular cache-cooperating scheme to increase the total client population. To study the performance of other cache-cooperating schemes in comparison to hierarchical caching, see [23]. Caches are usually placed at the access points between two different networks to reduce the cost of transmitting across a new network. As shown in Figure 1, we make this assumption for all three levels. Every institutional ISP has an institutional cache on level 1 of the tree, every regional ISP has a regional cache on level H, and every national ISP has a national cache placed on level $2H+1$. Therefore, there is one national cache, O^H regional caches, and O^{2H} institutional caches. Hierarchical caching works as follows. At the bottom level of the hierarchy are the client caches. When a document request is not satisfied by the client's cache, the request is redirected to the institutional cache. If the document is not found at the institutional cache, the request is then forwarded to the regional cache, which in turn forwards unsatisfied requests to the national cache. If a requested document is not found in the cache hierarchy the national cache requests the document directly from the 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.

3.3 Satellite Distribution

Given the model of the Internet described on Figure 2, we now consider the situation where the O^{2H} institutional caches cooperate via a satellite distribution (Figure 3). Every institutional ISP has an institutional cache with Internet connection, high storage capacity, and a satellite dish for receiving. Note that there is no more a caching hierarchy (i.e. regional or national caches). In addition to the institutional caches, there is also a master site which has an Internet connection and a satellite transmitter.

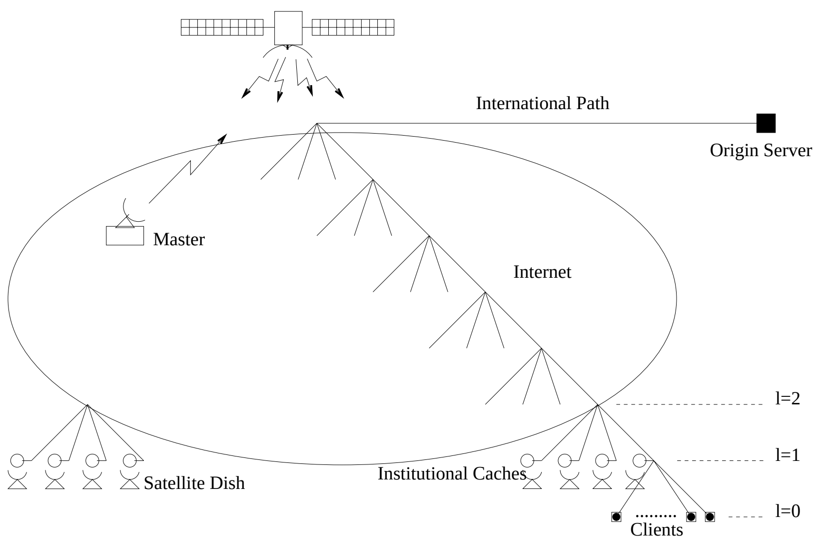


Figure 3: Cache-satellite distribution.

3.4 Web Documents

We denote the total number of documents in the WWW in the year j-th as N_j. Let x be the annual rate at which the number of documents in the WWW is increasing.

$$N_j = N_{j-1} \cdot x.$$

Denote S_i, the size in bytes for Web document i, $1 \leq i \leq N_j$ (we consider a document to be a Web page or an in-lined image). Web document i expires after an elapsed update interval t following an exponential distribution with average update interval Δ_i [11].

$$f\{t\} = \frac{1}{\Delta_i} e^{-t/\Delta_i}.$$

Let R_i be the number of requests for document i. Requests from an institutional ISP for document i are Poisson distributed with average request rate $\lambda_{I,i}$ [15].

$$P\{R_i = r|t\} = \frac{e^{-\lambda_{I,i}\cdot t} \cdot (\lambda_{I,i} \cdot t)^r}{r!}.$$

Assuming that requests for document i are uniformly distributed among all O^{2H} institutional caches, there are $\lambda_{I,i} \cdot O^{2H}$ total requests for document i. Let $\beta_{I,j}$ be the request rate from an institutional ISP for all N_j documents, $\beta_{I,j} = \sum_{i=1}^{N_j} \lambda_{I,i}$. $\beta_{I,j}$ is Zipf distributed [6] [33], that is, if we rank all N_j documents in the Web in order of their popularity, the $i-th$ most popular document has a request rate $\lambda_{I,i}$ given by

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

where α takes values between 0.6 and 0.8 [6], and σ is given by $\sigma = (\sum_{i=1}^{N_j} \frac{1}{i^{\alpha}})^{-1}$.

We assume that each document is requested independently from other documents, so we are neglecting any source of correlation between requests of different documents. We consider that newly appearing documents will also be Zipf distributed. That is, there will be some new appearing documents that will be very popular but there will also be many other new documents that will be requested by few clients. Let θ be the percentage of requests that are for non-cacheable documents (private documents, cgi-bin, etc).

4 Performance Analysis

In this section we present analytical models to calculate upper bounds on the performance of a cache-satellite distribution. We derive expressions to calculate the achievable hit rate at an institutional cache connected to the satellite distribution and at an institutional cache not connected to the satellite distribution. We also derive simple expressions to calculate the bandwidth needed for the satellite link, the disk space requirements at the caches, and the document-access latency. The goal of this section is not to exactly model empirical results, but to examine the impact and sensitivity of a large range of parameters in a cache-satellite distribution.

4.1 Hit Rate

The hit rate is the percentage of requests that find an up-to-date version of the requested document in the cache. We assume that the caches have an infinite capacity and therefore documents are not removed from the cache due to storage limitations. First, we analyze the hit rate at an institutional cache not connected to the satellite distribution and then we consider the hit rate at an institutional cache connected to the satellite distribution.

Let L be a random variable denoting the number of links traversed to hit a document. $P\{L = 1\}$ is the probability to meet a document at the institutional cache. The steady-state hit rate Hit_j in an institutional cache in year j, is given by

$$Hit_j = \sum_{i=1}^{N_j} \frac{\lambda_{I,i}}{\beta_{I,j}} \cdot P\{L = 1\} \cdot (1 - \theta) \quad (1)$$

where $P\{L = 1\}$ is given by

$$P\{L = 1\} = \int_0^{\infty} P\{L = 1|t\} \cdot f\{t\} \cdot dt \quad (2)$$

Let τ denote the time into the interval $[0, t]$ at which a request for document i occurs. The random variable τ is uniformly distributed over the interval $[0, t]$. Thus,

$$P\{L = 1|t\} = \frac{1}{t} \cdot \int_0^{t} P\{R_i > 0|\tau\} \cdot d\tau. \quad (3)$$

where $P\{R_i > 0|\tau\}$ is the probability that there has been at least one request for document i in the interval $[0, \tau]$. For an institutional cache not connected to the satellite distribution $P\{R_i > 0|\tau\} = 1 - e^{-\lambda_{I,i} \cdot \tau}$. Evaluating $P\{L = 1|t\}$ we obtain

$$P\{L = 1|t\} = 1 + \frac{e^{-\lambda_{I,i} \cdot t} - 1}{\lambda_{I,i} \cdot t}. \qquad (4)$$

Combining equations(2) and (4) we obtain

$$P\{L = 1\} = 1 + \frac{1}{\lambda_{I,i} \cdot \Delta_i} \cdot ln(\frac{1}{1 + \lambda_{I,i} \cdot \Delta_i})$$

When all institutional cache are connected to the satellite distribution, only the first request for document i out of all requests for the same document in all institutional ISPs sees a document miss. The rest of the requests will see a document hit. The hit rate at any institutional cache when O^{2H} institutional caches are connected to the satellite distribution can be calculated in the same way than for no cache-satellite distribution, using an effective client request rate equal to $\lambda_{I,i} \cdot O^{2H}$. Since the effective client population is larger, the probability that there has been already at least one request for document i by the time a client requests document i is given as $P\{R_i > 0|\tau\} = 1 - e^{-\lambda_{I,i} O^{2H} \tau}$, and is close to one.

4.2 Disk Space

To quantify the disk requirements of the caches, we calculate the cache disk space D_j in year j. Caches need to keep a copy of every document distributed through the satellite during the period of time that the document is up-to-date. Since documents expire after t time units, a cache needs to keep all documents that have at least one request during a period t. The required disk size $D_j(t)$ in a cache for a given a period t, is

$$D_j(t) = S \cdot \sum_{i=1}^{N_j} \cdot (1 - e^{-\lambda_{I,i} O^{2H} t}),$$

where $(1 - e^{-\lambda_{I,i} O^{2H} t})$ is the probability that document i has at least one request during a period t. Thus, the average disk space D_j in year j is given by $D_j = \int_0^{\infty} D_j(t) \cdot f\{t\} dt$.

4.3 Bandwidth

Now we calculate the bandwidth needed for the satellite link to transmit all newly appearing documents as well as all document updates. The bandwidth BW_j needed in year j, is given by

$$BW_j = \sum_{i=1}^{N_j} bw_i \qquad (5)$$

where bw_i is the bandwidth needed to transmit document i

$$bw_i = \int_0^{\infty} bw_i(t) \cdot f\{t\} dt \qquad (6)$$

and $bw_i(t)$ is the bandwidth needed to transmit document i given an update period t. $bw_i(t)$ is given by

$$bw_i(t) = \frac{S_i}{t} \cdot (1 - e^{-\lambda_{I,i} O^{2H} t}) \qquad (7)$$

where $1 - e^{-\lambda_{I,i} O^{2H} t}$ is the probability that document i is requested at least once in the period t from any institutional ISP.

4.4 Latency Analysis

Given the hierarchical topology of the Internet with very congested top levels and much less congested lower levels, documents found at low network levels close to the clients experience small latencies. To quantify the latency we calculate the expected number of network links that a request travels to hit a document. We analyze the latency in the case that institutional caches cooperate through a caching hierarchy and in the case that institutional caches are connected through a satellite distribution.

Let L_n be the number of links traversed by the $n-$th request for the same version of a certain document. The average number of links $E[L_n]$ traversed by the n-th request, is given by

$$E[L_n] = \sum_{l \in \{1, H+1, 2H+1, 2H+z+1\}} l \cdot P\{L_n = l\}$$

We now calculate the distribution of L_n. The first request $n = 1$ for a document in an update interval always travels to the origin server, i.e., $P\{L_1 = 2H + z + 1\} = 1$. In the case that institutional caches cooperate through a caching hierarchy, for $n \geq 2$, $P\{L_n = l\}$ is given by

$$P\{L_n = l\} = \begin{cases} 1 - (1 - \frac{1}{O^{2H}})^{n-1} & l = 1 \\ (1 - \frac{1}{O^{2H}})^{n-1} - (1 - \frac{1}{O^{H}})^{n-1} & l = H + 1 \\ (1 - \frac{1}{O^{H}})^{n-1} & l = 2H + z + 1 \end{cases} \tag{8}$$

In the case that institutional caches are connected through a satellite distribution we have $P\{L_n = 1\} = 1$, for $n \geq 2$.

5 Numerical Results

In this section we pick some reasonable values for the different parameters used in the analytical models to obtain upper bounds on the requirements and performance of a cache satellite distribution. These values will be fix for the rest of the paper unless stated differently.

The number of documents N_j in the Web at the beginning of the year j=1998 is about 250 millions [5]. The Web is increasing at a rate such that the number of documents get duplicated every year ($x = 2$) [5]. We consider that the HTTP traffic generated in all the caches connected to the satellite distribution ($O^{2H}\beta_{I,j}$) in $j = 1998$ is equal to $10,000$ document requests per second (this value is equal to the Internet backbone traffic in 1998 [7]). The HTTP traffic grows by a factor of 2.8 per year due to the increasing number of clients, the increasing period of time that clients are connected, and the increasing bandwidth of clients' connections [7].

We consider that there is no relationship between the update period of a document and its popularity [6]. Thus, we assume that all Web documents expire randomly with the same average update period $\Delta_i = \Delta$. We vary Δ from 1 hour to 20 days [11]. We also assume that the document size is independent of the document's popularity and the rate of change [6]. Therefore, we consider the same document size $S_i = S$ for every Web document, varying from 10 KB to 20 KB[19].

5.1 Latency Analysis

To consider real values for the latency, we analyzed 10 days of logs for the local proxy at EURECOM, which is connected to a 3-level caching hierarchy. We averaged the latencies during the 10 days of the trace to obtain the following values: transmission time from the local cache= 117 msec, transmission time from the regional cache= 550 msec, transmission time from the national cache= 800 msec, transmission time from the origin server= 1183 msec. Combining the probability $P\{L_n\}$ (equation 8) that the n-th request hits a document at level L, and the average latency to retrieve a document from network level L, we can calculate the average latency experienced by the n-th client request for a generic document.

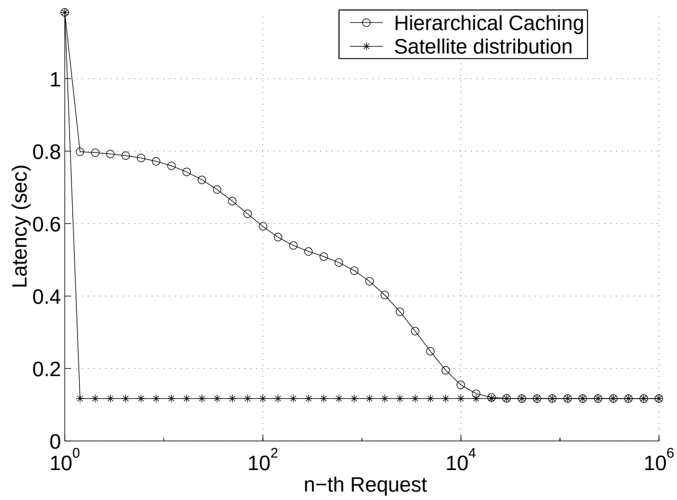


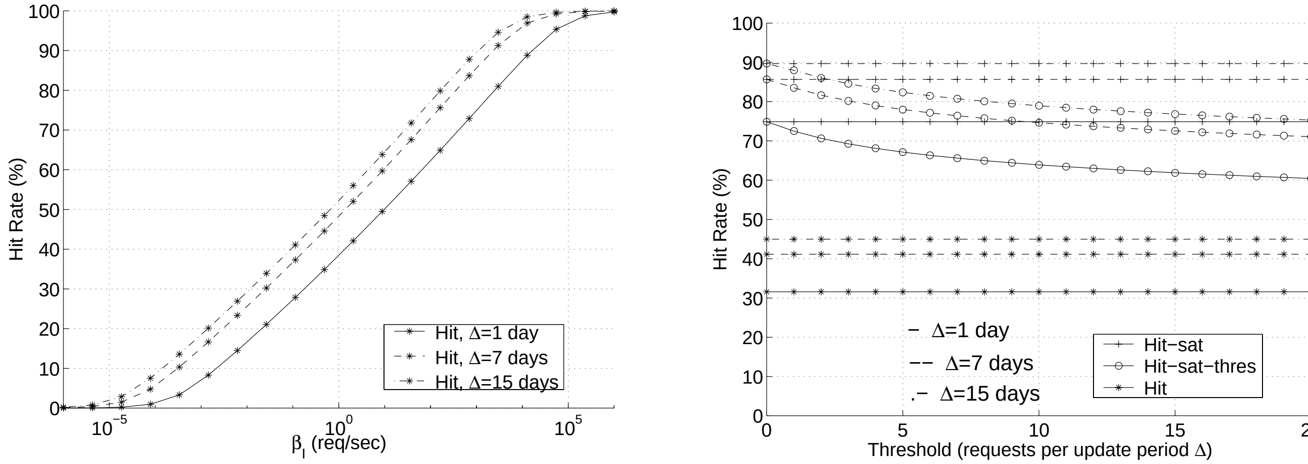
Figure 4: Average latency experienced by the n-th client in a 3-level hierarchical caching distribution and in a cache-satellite distribution. $O = 4$, $H = 3$, $z = 1$.

In Figure 4 we considered the latency experienced by a client in a three level caching hierarchy and in a cache satellite distribution. In both schemes, we see that the first request for a document always needs to travel to the origin server, thus experiencing high latencies ($1, 183$ msec). In hierarchical caching, the probability to find a document at closer to the clients increases slowly as the document becomes more and more popular. Thus, it is necessary to have a very high number of requests per update period ($n > 10^4$) for clients to experience average latencies equal to transmission time from the institutional cache. In a cache-satellite distribution, on the other hand, the document is pushed to all institutional caches after the first client request. Thus, every client request after the first one ($n > 1$), is directly satisfied from the institutional caches. As a result, a cache satellite distribution clearly reduces the latency to the clients by bringing documents into edge caches after the first document request.

5.2 Hit Rate

Figure 5(a) shows the hit rate for cacheable documents as a function of the total request rate (β_I) at a single cache (the population size can be easily calculated by assuming a certain client request rate, e.g. 500 requests/day [2]). We observe that for caches with small request rates, the hit rate is very small since the sharing among clients is very limited. As the number of requests increases, the hit rate increases with the logarithm of the request rate. However, it is necessary for a cache to have at least 100 to 1000 requests per second to satisfy 80% to 90% of the cacheable requests. This request rate is equivalent to having a population size of about $17, 300$ to $173, 000$ clients connected to the cache, which is a very large value for many ISPs. As a result, ISPs with small client populations usually deploy inter-cache cooperation protocols with other ISPs at the same or different network levels to increase the effective client population and improve their hit ratios.

A satellite distribution, on the other hand, can easily increase the effective client population connected to the cache with no inter-cache communication protocol. In Figure 5(b) we considered the case of a single cache with $\beta_I = 15$ requests per second, and the case of $O^{2H} = 64$ identical caches that cooperate using a cache-satellite distribution



(a) Hit rate for a single cache as a function of the request rate. (Log scale on x-axis).

(b) Hit rate for $O^{2H} = 64$ caches connected to the satellite distribution, and for a single cache, as a function of the popularity threshold.

Figure 5: Hit rate analysis ($\%$ of cacheable requests).

and generate 960 requests per second. Again we plot the hit rate for cacheable objects. Our analytical results for the hit rate on a single cache not connected to the satellite distribution indicates values ranging from 33% to 45%, which are very similar to the actual hit rates reported for many institutional caches [3]. For a cache connected to the satellite distribution we obtain hit rates from 80% to 90%. Thus, when individual caches get inter-connected through a cache-satellite distribution, the hit rate doubles.

Given the long-tailed distribution of Web documents, there are many documents that are requested only once or less in an update period. Documents requested only once in an update period are not shared by several clients and should not be cached. Therefore, the master site can take the decision to broadcast only those documents that have a number of requests per update period Δ higher than a certain *threshold*. In Figure 5(b) we also show how the hit rate for cacheable documents varies as a function of the threshold. For small thresholds and a satellite distribution, the cacheable hit rate is smaller than 100% since there are many documents that are requested only once during an update period. As the threshold value increases, the hit rate decreases, however, even for very high threshold values, e.g. 20 requests per update period, the hit rate is not decreased by more than 15%. As we will see in next section, this is a very important result since using threshold filtering policies allows caches to achieve high hit rates while reducing the disk requirements and bandwidth needed for the satellite link.

5.3 Caches Capacity

The price of disks is decreasing faster than the price of the network capacity. Additionally, the capacity of the clients' disks is increasing at a rate of 60% per year [7] with a baseline of 9 GBytes in 1998 (Figure 6). In a few years it will be easy to find disks with capacities close to TBytes at current prices [16].

With such large storage capacities it will be feasible to store a large portion of the Web on several disks at different points in the network. In figure 7(a) we show the disk capacity needed to store all Web documents that have at least one request per update period ($R_i > 0$). The values presented are an an upper bound for the storage requirements in a cache. Taking an average document size of $S = 10$ Kbytes and $S = 20$ Kbytes, the total number of documents distributed through the satellite can be stored in several TBytes (i.e., 20 TBytes in the year 2000). These results match very well those reported in a recent study [19], where the size of the publicly indexable Web pages and

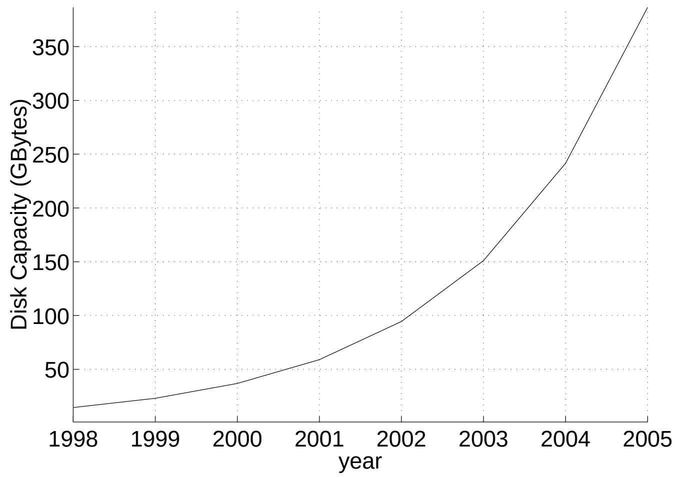
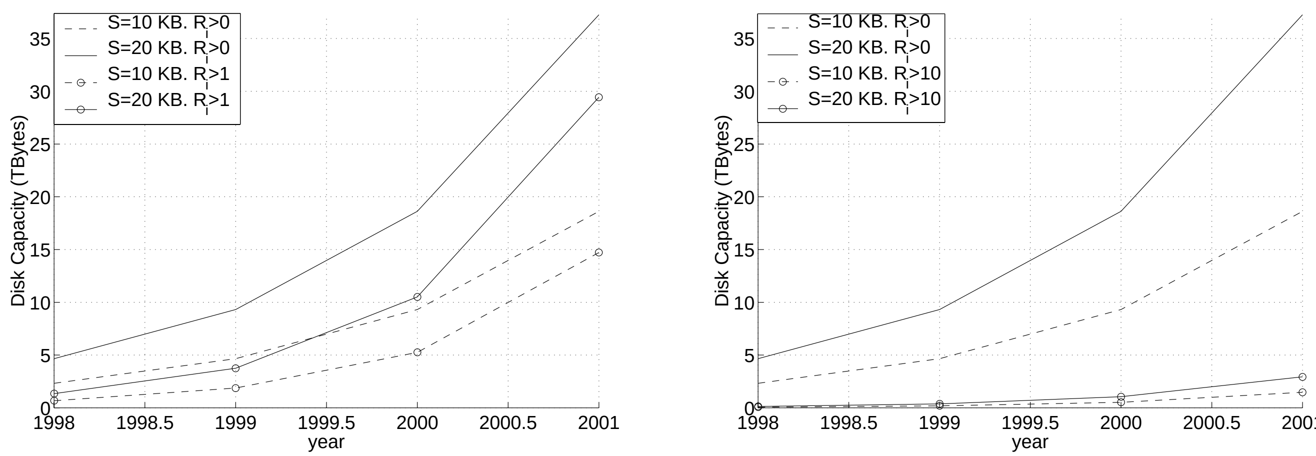


Figure 6: Disk Capacity Trend.

images in the middle of 1999, was estimated to be 18 TBytes. The storage capacity to store all newly appearing Web documents needs to be doubled every year, following the rate at which the number of Web documents is increasing.



(a) More than once.

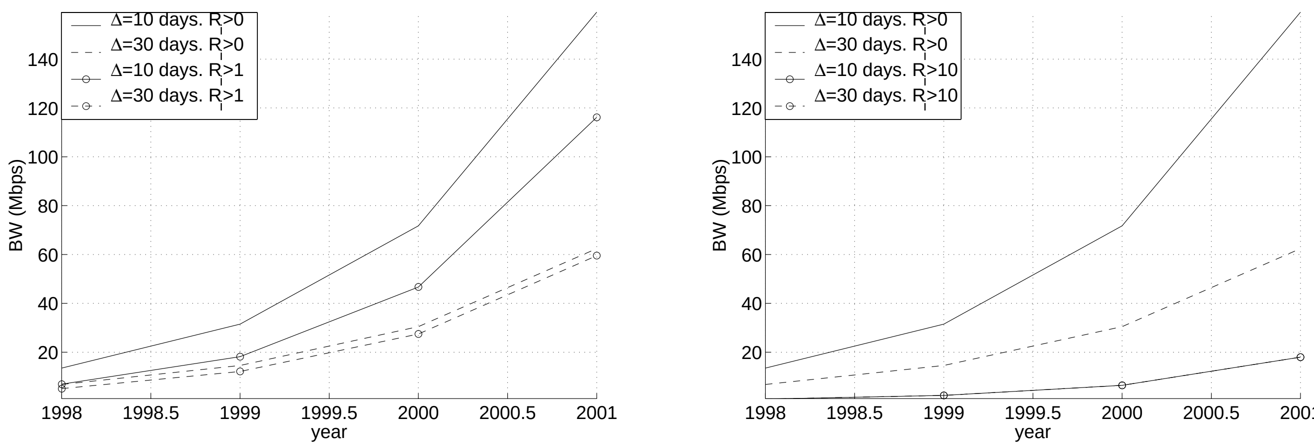
(b) More than ten times.

Figure 7: Disk capacity needed to store all Web documents send through the satellite.

In Figure 7(a) we see that if the master side only transmits those documents requested more than once ($R_i > 1$), the needed storage capacity drops to 1.8 TBytes in year 1999 and increases at rate equal to rate at which the HTTP Internet backbone traffic increases (2.8 times per year). In Figure 7(b) we see that if the master site only transmits those documents requested more than 10 times, the needed storage capacity in the year 1999 drops even more to 187 Gbytes and increases at a much smaller rate. Caches with storage capacities of several hundreds of Gbytes are already common today [25]. This simple filtering policy saves a lot of disk space in the institutional caches and reduces the hit rate at an institutional cache by less than 10% (see previous Section). In the case that caches do not have such storage capacities, more sophisticated filtering policies could be implemented at the client side to reduce the needed storage capacity while hardly affecting the hit rate [17].

5.4 Bandwidth

As we discussed in Section 4.3, the satellite needs to send document updates for existing documents and newly appearing documents in the Web. Using equation 5, we plot in Figure 8 the necessary bandwidth for the satellite link to keep broadcasting newly appearing documents and document updates that are requested at least once in an update interval ($R_i > 0$). We take the size of a Web document as $S = 10$ and vary the update period Δ from 10 to 30 days. From Figure 8(a) we can observe that the bandwidth needed to send documents through the satellite is close to 31 Mbps for year 1999 and that keeps increasing with a rate close to 2 per year. The rate at which the bandwidth increases depends on the rate at which the HTTP Internet backbone traffic increases and on the rate at which new documents appear in the Web. If all appearing documents were requested at least once in an update period, the bandwidth would increase at the same rate at which Web documents appear in the Web (factor of 2 per year).



(a) More than once.

(b) More than ten times.

Figure 8: Required bandwidth at the satellite to transmit new documents and document updates. $S = 10$ KBytes.

As discussed in Section 5.3, there is a large group of documents that are only requested few times in an update interval. From Figure 8(a) we see that when the master site only transmits documents requested more than once ($R_i > 1$) the bandwidth needed is reduced to 19 Mbps for the case of $\Delta = 10$ days, and year 1999. When the update interval Δ increases, new updates must be distributed less frequently and the bandwidth needed decreases, i.e. for $R_i > 1$, $\Delta = 10$ days, and year 1999 the bandwidth needed for the satellite link drops to 12 Mbps. A satellite distribution of an analog television channel uses a bandwidth of 27 Mbps. Thus, with the capacity needed to broadcast one TV channel a cache-satellite distribution can be used to distribute almost all Web documents in the year 1999.

Figure 8(b) shows the bandwidth needed at the satellite link when the master site only transmits those documents requested more than 10 times. In this case, the needed bandwidth at the satellite in 1999 is equal to 2.2 MBps. These numbers approximate those provided by the cache-satellite company Sky-Cache [27], which in 1998 was using a 4 Mbps link to feed Web documents into the subscribed caches and only broadcasted those documents that were requested several times during an update interval (e.g., more than two or three requests per update interval).

6 Trace-Driven Simulations

In this section we perform a trace-driven analysis to evaluate the hit rate of an institutional cache that it is not connected to a cache-satellite distribution and the hit rate of an institutional cache connected to a cache-satellite distribution. We also use the trace driven results to validate the analytical model developed in Section 4.

First, we will present the distribution of hits and misses in an institutional cache to better understand the potential benefits of a cache-satellite distribution.

6.1 General Distribution of Hits and Misses in a Single Cache

To investigate the distribution of hits and misses, we analyzed the logs of our institutional cache at Institut EURECOM [2]. We took 7 days of logs from the 10th of December to the 17th of December of 1998, which include $187,054$ requests. Our institutional cache has 6 GBytes of disk space and connects about 100 clients.

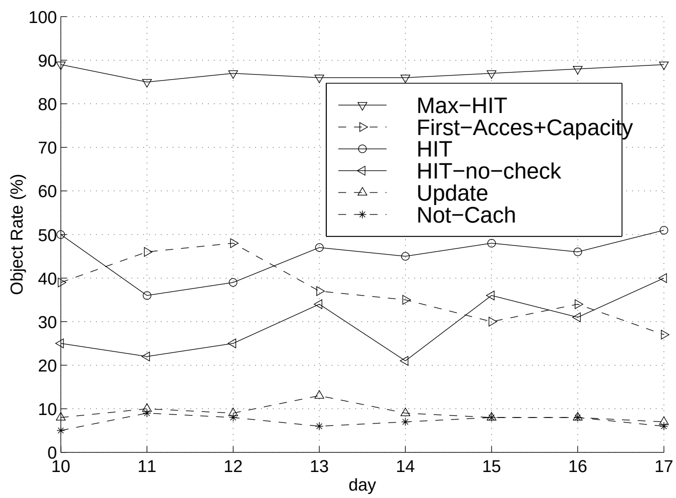


Figure 9: General distribution of EURECOM's cache entries. Hits are shown with a continuous line, misses are shown with a dashed line.

Figure 9 shows the distribution of the object rate for the different days of the logs (results for the byte distribution show a similar behavior). The total hit rate (*HIT*) includes documents that are directly satisfied from the cache and documents that are satisfied from the cache but need a previous check with the origin server. We see that the total *HIT* ranges from 35% to 50%. We see that the hit rate for documents for which the cache does not need to contact the origin server (*HIT-no-check*) ranges from 20% to 35%. The percentage of non-cacheable objects (*Not-Cach*) never exceeds 10%. We consider a document to be uncacheable if 1) the document was created by a cgi-bin script, 2) the document requires authorization, 3) the document has an HTTP 1.1 cache-control header to explicitly mark it as non-cacheable, 4) the document request is a query, 5) the document response has a "pragma: no cache" header, or 5) the document request method is other than GET. We did not consider documents carrying a Set-Cookie header as uncacheable, since the HTTP 1.1 protocol, as opposed to the HTTP 1.0 protocol, allows these responses to be cached. The percentage of uncacheable objects that we have found in EURECOM's cache has also been confirmed by other studies [28] [22]. The sum of first-access misses, capacity misses, and update misses (*First-Access+Capacity+Update*) account for as much as 50% of all requests. We expect capacity misses to be a very small fraction of these requests since EURECOM's cache has a 6 GB disk space, which is enough for EURECOM's small client population [25]. The percentage of update misses (*Update*) is about 9%, therefore, the first-access misses account for a big percentage of the misses (30%-50%). The number of first-access misses is very high is because EURECOM's cache has a very small and diverse client population.

Next, we calculate the *maximum achievable hit rate (Max-HIT)* assuming an ideal scenario where institutional caches have an infinite storage capacity and all documents are prefetched in the cache before they are requested. In this ideal case, all requests are hit in the institutional cache, except requests for non-cacheable documents and requests where the client imposes a poll check with the origin server (i.e., reload). This scenario is equivalent to having a huge number of institutional caches cooperating via a satellite distribution where the probability that a client is the first client asking for a document is very small and thus first-access misses, and update misses become hits. The *Max-HIT* at EURECOM's cache is about 85%. Therefore, if EURECOM's cache would be connected to a cache-satellite distribution with many cooperating institutional caches the hit rate could improve by a maximum of 30% to 50%.

6.2 Cache-Satellite Distribution Study

To investigate the effect of a cache-satellite distribution we analyzed the logs of an ISP in the USA (AYE [1]), which is connected to a real cache-satellite distribution [27]. We took one week of logs (Dec 18-25, 1998), which account for $13,689,620$ requests. This ISP gives access to about 1000 residential and business clients and has an institutional cache with 48 GBytes of disk space that is connected to the SkyCache [27] satellite distribution. SkyCache's cache-satellite distribution was joining at the time the logs were collected more than 30 ISPs in the USA, including several national ISPs. SkyCache was pushing new documents and document updates at a rate of 4 Mpbs twenty four hours a day. The master center of the SkyCache satellite distribution was only broadcasting those documents that received several requests in an update period, depending on the available satellite bandwidth [27].

Analyzing the log traces from AYE's cache we could identify hits produced by documents pushed through the satellite channel, and hits produced by documents previously fetched by local clients, which are independent of the cache-satellite distribution. Thus, we could calculate the hit improvement in AYE's cache achieved by the cache-satellite distribution. Figure 10 shows that the total *HIT* rate at AYE's cache if the cache would not be connected to the satellite distribution is 30%-40%. The number of first-access misses is much smaller than in the EURECOM's trace, 25%-35%, since AYE's client population is much higher than EURECOM's. When AYE's cache is connected to the satellite distribution there are many requests that are satisfied by documents previously pushed through the satellite distribution. The increase in the hit rate offered by a cache-satellite distribution in AYE's cache is about 18%-25%, thus, AYE's cache is having a total local hit rate (*HIT+Satellite*) of about 60%. The maximum possible hit rate (*Max-HIT*) assuming the idealized case where all documents are prefetched into the institutional caches and caches have infinite disk is about 90%.

6.3 Scaled Satellite Distribution

Now, we proceed to simulate the case where the cache-satellite distribution network of which AYE's cache is part, has a much larger client population. For this purpose we took seven days of logs from the four major parent caches, "bo", "pb", "sd", and "uc" in the National Web Cache hierarchy of the National Lab of Applied Network Research (NLANR) [3] from Dec 18th to 25th, 1998 ($32,892,307$ requests). These caches are the top-level caches of the NLANR caching hierarchy and all cooperate to share their load. We simulate the scenario where we have the satellite distribution described in the previous section and then we assume that NLANR top-level caches get also connected to the satellite distribution.

To simulate this new scenario, we consider that AYE's cache is connected to the SkyCache network. Then, for every request in the AYE's trace that results in a miss, we check if the request could be satisfied by any of the documents requested in the NLANR trace. If some client in the NLANR caches previously accessed the document missed in AYE's cache, the document would have been distributed through the satellite and would result in a local hit in AYE's cache. To model a more realistic situation, we consider that the AYE cache is finite and can only keep a certain number of documents pushed through the satellite from NLANR (i.e., one day). This corresponds to a storage capacity of 16 Gbytes, which have to be added to the 48 GB that AYE's cache already had.

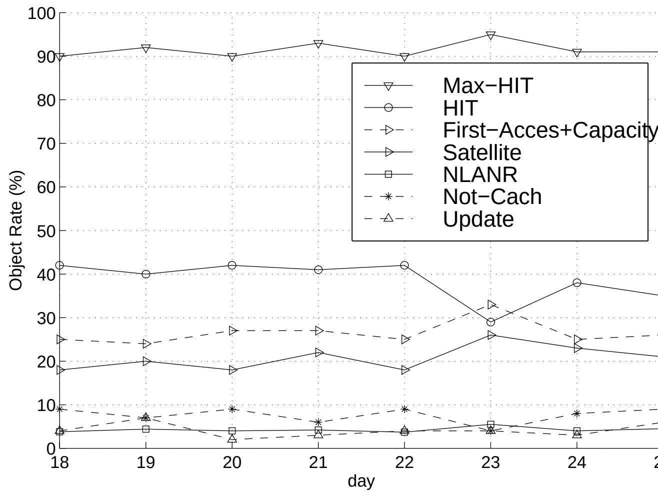


Figure 10: General distribution of AYE's proxy log entries. Hits are shown with a continuous line, Misses are shown with a dashed line. NLANR entry shows the additional hit rate achieved when NLANR top-level caches also connect to the satellite distribution.

In the absence of MIME headers (i.e., last-modified, expires, etc.) in both, NLANR or AYE traces, it is difficult to ensure that a document miss in AYE's cache, whose URL matches one entry in the NLANR logs, is still up-to-date. To verify if two document requests with the same URL correspond to the same document version, we compared the documents size of the request in the AYE logs and in the NLANR logs. Thus, if the URL and the document size of a miss in the AYE logs matches the URL and the document size of any entry during the previous 24 hours of the NLANR logs, we account for a new document hit.

In Figure 10 we show the increase in the hit rate at AYEs institutional cache when NLANR caches get also connected to the satellite distribution. The additional hit rate offered by the NLANR traces is 4% to 5%, resulting in a total hit rate (*HIT+Satellite+NLANR*) of about 65%. The benefits achieved by adding the NLANR logs are not very high since AYE's cache is already connected to a cache-satellite distribution with a large effective client population. When the number of caches connected by the satellite is already large, adding a new cache to the cache-satellite distribution offers only minor improvements. In the same way, when ISPs with a large client population connect to the cache-satellite distribution, the expected hit rate improvement is small. However, the satellite distribution helps reducing the bandwidth costs of any ISP, since documents are prefetched into the caches using the satellite link, freeing the expensive Internet terrestrial connections.

Hit rates close to 65% have also been reported in a caching hierarchy [25][28]. However, in a caching hierarchy a request needs to travel up in the network through a series of caches until the document is hit. These caches can be very congested or can be connected trough very congested links, in which case clients can experience high response times. Instead, a cache-satellite distribution can achieve the same hit rates locally with fast response times, low bandwidth usage, and a minimum configurations. However, there is still a big gap between the 65% hit rate and the maximum achievable hit rate 90%. This can be due to the fact that the local cache is only preloaded with one day of logs from the NLANR caches; if more days would be used the hit rates would increase. Also, we should note that a satellite distribution needs to receive the first request for a document before it is transmitted through the satellite. If the number of documents that are requested only once is high, the maximum achievable hit rate is limited (see Section 5.2).

7 Conclusions

Caching is being extensively deployed in the Internet to alleviate the problems related to the exponential growth of the Web. However, caching has a limited performance due to the high number of requests not hit in the cache. Requests not hit in a cache are requests for documents that have been purged from the cache, requests for non-cacheable documents, or requests for documents that no client has fetched before. Since the price of the disks is dropping very fast and the capacity of the disks is increasing at very high rates, we expect that only few documents will be purged from the caches due to space constraints. Only when the number of multimedia documents and continuous media streams grows, limited disk capacity may become a problem again. Non-cacheable documents may become cacheable if some intelligence is placed in the caches to cope with dynamic content [8]. To reduce the number of requests that result in first requests for a document, a cache needs to be shared by a large client population.

In this paper we analyzed a cache-satellite distribution that interconnects many caches and creates an effective client population that is very large. A cache satellite distribution does not require any inter-cache cooperating protocol and drastically reduces the number of requests that are first requests for a certain document. We have presented analytical models and performed trace-driven simulations to evaluate the performance and requirements of a cache-satellite distribution in terms of hit rate, disk space at the caches, document-retrieval latency, and bandwidth needed at the satellite. We have found that small ISPs with a cache disk space of about 64 GB connected to a cache-satellite distribution, can improve their local hit ratios by a factor of 25% to 35%. For ISPs with a large client population (e.g., several thousand clients), using a cache-satellite distribution offers only marginal improvements on the hit rate. However, using a cache-satellite distribution ISPs can significantly reduce the cost of filling their caches, freeing up more expensive Internet terrestrial links for other services.

In the case that home clients are also connected to the satellite distribution, they do not suffer the last mile problem for the Web traffic since documents are pushed directly to their disks. As a last remark, one should notice that if the percentage of uncacheable documents increases, the importance of a cache-satellite distribution or any other cache-sharing protocol will be less relevant and a higher effort should be made to increase document's cacheability.

8 Acknowledgments

The support of P. Rodriguez by the European Commission in form of a TMR (Training and Mobility for Researchers in Europe) fellowship (ERBFMBICT972639) is gratefully acknowledged. We would like to thank Keith W. Ross for his help and fruitful discussions, Peter Cohen from SkyCache, and Jim Hardley from AYE for his cooperation in collecting the traces. Thanks to NLANR for making their logs available. EURECOM's research is partially supported by its industrial partners: Ascom, Cegetel, France Telecom, Hitachi, IBM France, Motorola, Swisscom, Texas Instruments, and Thomson CSF.

9 Vitae

Pablo Rodriguez Pablo Rodriguez (rodrigue@eurecom.fr) obtained his M.S. degree as a Telecommunication Engineer from the University of Navarra UPNA, Spain in 1995. He did his Master Thesis at the department of Electrical and Electronic Engineering at King's College, London. Then, he moved to the Swiss Federal Institute of Technology EPFL, Lausanne where he did Postgraduate Studies in Communication Systems. Since September 1997 he is pursuing a Ph.D. at Institut EURECOM, France with a TMR (Marie Curie) fellowship. From June 1999, until October 1999 he was at AT&T Labs Research, New York.

His research interests are large scale document dissemination, reliable multicast transmission, caching schemes for the Web and push technologies.

Ernst W. Biersack Ernst Biersack (erbi@eurecom.fr) received his M.S. and Ph.D. degrees in Computer Science from the Technische Universität München, Munich, Germany.

Since March 1992 he has been a Professor in Telecommunications at Institut Eurecom, in Sophia Antipolis, France. For his work on synchronization in video servers he received in 1996 (together with W. Geyer) the Outstanding Paper Award of the IEEE Conference on Multimedia Computing & Systems. For his work on reliable multicast he received (together with J. Nonnenmacher and D. Towsley) the 1999 W. R. Bennet Price of the IEEE for the best original paper published 1998 in the ACM/IEEE Transactions on Networking.

Mr. Biersack is currently an associate editor of IEEE Network Magazine, ACM/IEEE Transactions on Networking, and of ACM Multimedia Systems.

References

[1] "AYE Net, USA: http://www.aye.net/".

[2] "Institut EURECOM, Sophia Antipolis, France: http://www.eurecom.fr/".

[3] "National Lab of Applied Network Research (NLANR)", http://ircache.nlanr.net/.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] P. Cao, J. Zhang, and K. Beach, "Active Cache: Caching Dynamic Contents on the Web", In *Proceedings of IFIP*, pp. 373–388, Middleware, 1998.

[9] A. Chankhunthod, P. Danzig, C. Needaeles, and M. S. K. Worrell, "A hierarchical Internet object cache", In *Proceedings of 1996 Usenix Technical Conference*.

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

[11] F. Douglis, A. Feldmann, B. Krishnamurthy, and J.Mogul, "Rate of change and other metrics: A live study of the World Wide Web", In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, December 1997.

[12] 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.

[13] L. Fan, P. Cao, and Q. Jacobson, "Web Prefetching Between Low-Bandwidth Clients and Proxies: Potential and Performance", In *3rd International WWW Caching Workshop*, June 1998.

[14] 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.

[15] S. Gribble and E. Brewer, “System Design Issues for Internet Middleware Services: Deductions from a Large Client Trace”, In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, December 1997.

[16] M. L. Gullickson, A. L. Chervenak, and E. W. Zegura, “Using experience to Guide Web Server Selection”, , Georgia Institute of Technology, 1998.

[17] X.-Y. Hu, P. Rodriguez, and E. W. Biersack, “Performance Study of Satellite-Linked Web Caches and Filtering Policies”, Submitted to Networking 2000., EURECOM, Sep 1999.

[18] K. ichi Chinen, “An Interactive Prefetching Proxy Server for Improvements of WWW Latency.”, In *Proc. INET*, 1997.

[19] S. Lawrence and C. L. Giles, “Accesibility of Information on the Web”, In *Nature*, July 1999.

[20] T. S. Loon and V. Bharghavan, “Alleviating the latency and bandwidth problems in WWW browsing”, In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, December 1997.

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

[22] S. Manley and M. Seltzer, “Web Facts and Fantasy”, In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, December 1997.

[23] P. Rodriguez, C. Spanner, and E. W. Biersack, “Web Caching Architectures: Hierarchical and Distributed Caching”, In *Proceedings of the 4th International Caching Workshop*, San Diego, March 1999.

[24] K. W. Ross, “Distribution of Stored Information in the Web”, Online Tutorial, http://www.eurecom.fr/ ross/CacheTuto rial/DistTutorial.html.

[25] A. Rousskov, “On Performance of Caching Proxies”, In *ACM SIGMETRICS*, Madison, USA, September 1998.

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

[27] SkyCache, “http://www.skycache.com”.

[28] 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.

[29] 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.

[30] Z. Wang, “Cachemesh: A Distributed Cache System For World Wide Web”, , UCL, June 1997.

[31] 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.

[32] L. Zhang, S. Floyd, and V. Jacobson, “Adaptive Web Caching”, , University of California, Los Angeles, February 1997.

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