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

Pablo Rodriguez Ernst W. Biersack
Institut EURECOM
2229, route des Crêtes.
06904, Sophia Antipolis Cedex, FRANCE
{rodrigue, erbi}@eurecom.fr

September 30, 1998

Abstract

The World Wide Web is growing exponentially and already accounts for a big percentage of the traffic in the Internet. Popular Web servers are overloaded, hot documents travel many times across the same congested links, and receivers experience slow response times. Cache hit rates can be significantly increased by having caches cooperate. In this paper we analyze a scenario where caches cooperate via a push-satellite distribution [16] [6]. When a cache fetches a new document, the document is immediately broadcast to all other caches via the satellite. Caches connected to the satellite distribution end up containing all documents requested by a huge community of clients. The probability that a client is the first to request a document is very small. An ISP with a large cache connected to a push-satellite distribution can achieve bandwidth savings up to 90 %. Clients with local caches connected to the satellite distribution can alleviate the last mile problem and virtually browse the whole Web locally. We evaluate the feasibility of a satellite distribution to local caches and perform a quantitative analysis in terms of hit rate, latency, satellite bandwidth, and storage requirements. We also discuss how to schedule the push operation and present several techniques to filter non-desired documents.

Keywords: WWW, Caching, Push, Satellites

1 Introduction

The growth of the World Wide Web is overloading popular servers and increasing the traffic in the network and the response times to the clients. Web caching is being extensively used in the Internet to alleviate these problems. Web caching works as follows: when a client issues a request, the request is first directed to the cache. If the document is found in the cache the document is delivered directly to the client. If the document is not hit in the cache, the document is fetched from the origin server and a copy is placed in the cache. Further requests for the same document are satisfied from the cache. Requests not satisfied in the cache are called *misses*. Misses can be classified into:

- *First-Access*: Misses occurring when accessing documents the first time.
- *Storage Capacity*: Misses occurring when accessing documents previously requested but discarded from the cache to make space for other documents.
- *Updates*: Misses occurring when accessing documents previously requested but already expired.
- *Uncacheable*: Misses occurring when accessing documents that need to be delivered from the final server (e.g. dynamic documents generated from cgi-bin scripts or fast changing documents)

First-access misses are much higher than any other kind of misses [17] and may account for the 20% of all requests. In this paper we will focus on how to reduce the first-access misses as well as update misses. One way to reduce the number of misses is by sharing a cache with a large client population. The more users share the cache, the greater the chance that several clients are interested in the same document. One popular scheme to make caches cooperate is via a caching hierarchy [8][18]. In a caching hierarchy caches are placed at several levels of the network, including caches at the client side, at the metropolitan networks, at the regional networks, and at the national backbones. Caches cooperate at the same level of the caching hierarchy and at different levels of the caching hierarchy, sharing all the documents requested by their clients.

[*]In Proc. ACM/IEEE MobiCom'98 Workshop on Satellite-based Information Services (WOSBIS'98), Dallas

Another way to reduce the number of misses is by prepopulating the cache. Documents are pushed into the cache even if the cache has never fetched them. The idea is to get documents in the cache expecting that clients will likely request them later. Prepopulating caches has two main inconveniences: disk space and network bandwidth. If the cache keeps many documents that are never requested by its users a lot of disk space is wasted without any additional benefit. Disk space may not be such a problem because disk capacity is increasing at rate of 60% per year and large disks are becoming cheaper and cheaper [3][9]. However, if document sizes keep increasing and more multimedia documents are incorporated into Web documents, disk capacity may become limited. A more serious problem is network bandwidth, especially when caches are prepopulated via congested terrestrial links.

To scale a prepopulating distribution to many clients a multicast distribution should be used. Nevertheless, a multicast distribution in the Internet is not generally available and still needs further development. An alternative way to prefetch documents in the Internet is via a Caching hierarchy [15]. Documents are prefetched from one cache level to another cache level resembling a multicast distribution. The main drawback of any prefetch distribution over the Internet is the limited achieved throughput. If many documents are prefetched through congested links and no client requests them the bandwidth wasted is considerable.

An alternative way to feed caches with documents and therefore reduce the number of first-access misses is via a *satellite distribution*. A satellite distribution is being extensively deployed and bandwidth is plentiful. Companies willing to offer multicast applications are frustrated with the lack of multicast in the terrestrial network and are using satellite infrastructures which by default support multicast. A satellite distribution has fewer losses and congestion problems than a distribution in the Internet. Also, a satellite distribution can reach a very large population with relatively little effort; adding a new additional client does not increase the cost of transmission.

A satellite-push distribution to Web caches could work as follows. When a cache requests a document, the document is fetched at the origin server and kept locally. Additionally a small message is sent to a *master distribution center* that will get the document itself. Then, the master sends the document via satellite broadcast to all the other caches. Any other client that requests the same document, will hit the document in its cache. With this scheme a cache stores the documents requested by a much larger community, approximately the total number of users in all caches connected to the satellite distribution. Therefore, the probability that a single client is the first client asking for a document is very small.

There can be caches connected to the satellite distribution at any level of the network. As more caches join the satellite distribution the aggregation of clients is higher and the system works better. ISPs with large caches connected to the satellite distribution can achieve very high hit rates, obtain great bandwidth savings, and additionally increase their clients' satisfaction. Recently several companies like Sky Cache[16] and isp-sat [6] have started to offer this satellite distribution service.

A situation that deserves a special attention is the case where clients with local caches are also connected to the satellite distribution. A client receives all documents that are broadcasted by the satellite. If these clients do not have very large caches, they need to apply filtering policies to remove some of the documents. Clients benefit from local copies of the Web documents that best match their preferences achieving very high hit rates. They can surf these documents locally avoiding high response times due to congested links and slow modem connections.

In this paper we investigate the feasibility of a satellite-push scheme. We develop analytical models to quantify the achieved hit rate and the bandwidth needed to broadcast all documents through the satellite. We discuss several filtering policies at the master side as well as the at the client side. We describe how to schedule the push operation and compare a satellite-push distribution with an Internet distribution. We also present the economical implications of this scheme as well as introduce some possible future scenarios.

2 The Model

In this section we describe the main components of the system that will be the base of our analysis.

2.1 Hierarchical Caching in the Internet

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: metropolitan networks, regional networks, and national backbones. All of the clients are connected to the metropolitan networks; the metropolitan networks are connected to the regional networks; the regional networks are connected to the national networks.

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 metropolitan network has a metropolitan cache, every regional network has a regional cache, and every national network has a national cache. Additionally clients may also have local caches.

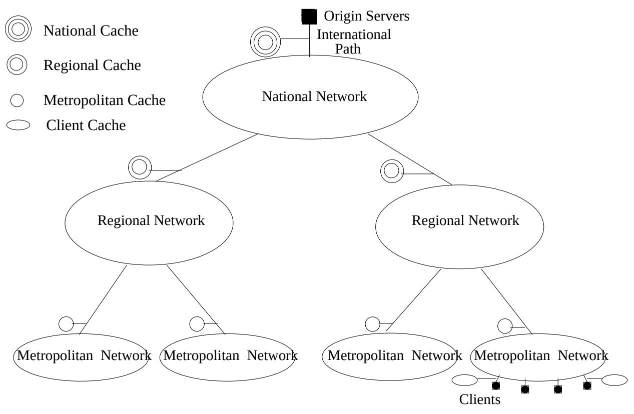

Figure 1: Internet topology

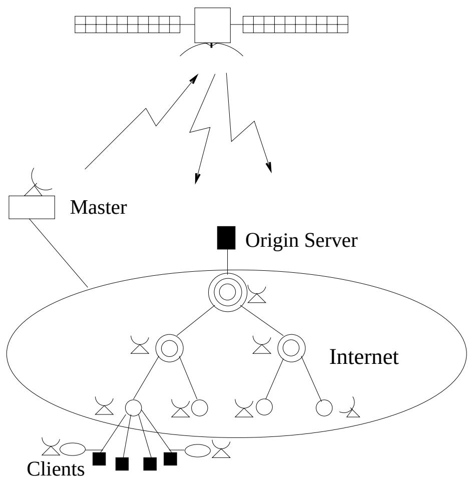

Figure 2: Push distribution from the servers to caches.

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 metropolitan cache. At the metropolitan level several caches may cooperate to increase the hit rate and distribute the load. If the document is not found at the metropolitan 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.

2.2 Satellite Distribution

Given the model of the Internet described on Figure 1, we now consider the situation where a satellite distribution is in place (Figure 2). The satellite receives documents from a master distribution center and forwards them to all caches. At any level of the network there can be caches connected to the satellite distribution. Caches receive all the documents requested by any client in the Web. After a certain period of time caches with high capacity could contain most of the documents in the Web. Only expired documents or new documents are not found in the cache. In the case that a cache is down when the satellite is broadcasting a certain document, the cache will miss this document. If the cache comes back and one of its clients requests that specific document the master will re-broadcast the document to all caches via the satellite. Caches receive the document twice and the bandwidth in the satellite is not used efficiently. One possible solution is for the master to check the document consistency before re-sending the document. If the document has not changed it is not retransmitted. If the document has changed it is retransmitted. Thus, the satellite sends a document at most once during the period where the document is up-to-date. To keep documents consistent, the master and the caches can use any of the current policies (adaptive TTL, poll every time) [11].

If a new cache joins the satellite distribution it may take several weeks for the cache to be filled. The cache will only receive new appearing documents and new versions of existing documents but not the rest of the documents that were already pushed. One way to solve this problem could be the following. The master distribution keeps a copy of all documents that were broadcasted and that are still up-to-date. When a new cache joins the system a copy of these documents can be shipped from the master center to the joining cache.

Satellite distribution of Web documents can be complementary to the current caching hierarchy using ICP [19]. If some of the caches in the caching hierarchy are not connected to the satellite distribution, they still can benefit from the documents pushed via the satellite into the neighbor caches. Additionally if a cache receives a request for a previously discarded document, it is very likely that the document is hit in the caching hierarchy before querying the origin server.

2.3 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 document i. Document i expires after an elapsed time T_i following an exponential distribution with average update interval Δ_i.

$$P(T_i = t) = \frac{1}{\Delta_i} e^{-t/\Delta_i}$$

Requests for document i are Poisson distributed with average request rate λ_i. Let $P(R_i = r|T = t)$ be the probability that the number of requests R_i for document i in an interval $T = t$ is equal to r.

$$P(R_i = r|T = t) = \frac{e^{-\lambda_i \cdot t} \cdot (\lambda_i \cdot t)^r}{r!}$$

The total number of requests for all N_j documents is also Poisson distributed with average request rate λ

$$\lambda = \sum_{i=1}^{N_j} \lambda_i$$

Recent studies show that the number of requests or all documents in the Web follows a Zipf distribution [5] [20]. Let all N_j documents in the Web be ranked in order of their popularity where document i is the i-th most popular document. The percentage of the total request rate that document i accounts for is given by:

$$\frac{\lambda_i}{\lambda} = \frac{\sigma}{i^\alpha}$$

where α is a parameter that usually takes values between 0.6 and 0.8 [5], 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 merge with existing documents into a new Zipf distribution. 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 (fast-changing documents, cgi-bin, etc).

3 Performance Analysis

In this section we present analytical models to calculate the hit rate in any cache connected to the satellite distribution, and the bandwidth required at the satellite.

3.1 Hit Rate

The hit rate is the percentage of requests that meet an up-to-date document in the cache. In a push-satellite distribution only the first request for a document will see a document miss. The rest of the requests for an up-to-date document see a document hit. For popular documents most of the requests will see a hit. However, for changing documents that are rarely requested most of the requests see a miss.

We analyze the hit rate in one cache depending on how fast documents change and depending on the number of documents in the Web. We assume that the caches have an infinite capacity and therefore documents are not removed from the cache due to storage limitations.

Let Hit_j be the hit rate given that there are N_j documents in the Web.

$$Hit_j = \sum_{i=1}^{N_j} \frac{\lambda_i}{\lambda} \cdot hit_i \cdot (1 - \theta) \qquad (1)$$

where hit_i is the hit rate for document i.

$$hit_i = \int_0^\infty hit(T_i = t) \cdot P(T_i = t) \cdot dt \qquad (2)$$

$$hit(T_i = t) \quad = \quad \frac{1}{t} \cdot \int_{\tau=0}^{\tau=t} P(R_i > 0|T = \tau) \cdot d\tau$$

where the hit rate $hit(T_i = t)$ in an update interval $T = t$ is calculated as the probability $1/t$ that a request for document i arrives at time $T = \tau$, $\tau \in [0, t]$, which is uniformly distributed, and the probability $P(R_i > 0|T = \tau)$ that there has been at least one request for document i before time $T = \tau$. Evaluating $hit(T_i = t)$ we obtain

$$hit_i(T_i = t) = 1 + \frac{e^{-\lambda_i \cdot t} - 1}{\lambda_i \cdot t} \qquad (3)$$

Combining equations(2) and (3) we obtain

$$hit_i \quad = \quad 1 + \frac{1}{\lambda_i \cdot \Delta_i} \cdot ln(\frac{1}{1 + \lambda_i \cdot \Delta_i})$$

3.2 Bandwidth

If Web documents would not change and new documents did not appear, after a certain period the satellite would not need to send any more documents and caches would contain all Web documents. However, Web documents change and new documents appear at very high rates. Thus, the satellite needs to keep transmitting all document updates and newly appearing documents. We calculate the necessary bandwidth at the satellite to distribute and to broadcast **all** new documents depending on how fast new documents are produced and how fast documents change.

The necessary bandwidth BW_j to broadcast all document updates, and new appearing documents during the year j is given by:

$$BW_j = \sum_{i=1}^{N_j} bw_i \qquad (4)$$

where bw_i is the bandwidth to broadcast updates for document i.

$$bw_i = \int_0^\infty bw(T_i = t) \cdot P(T_i = t) dt \qquad (5)$$

$$bw(T_i = t) = \frac{S_i}{t} \cdot (1 - exp(-\lambda_i \cdot t)) \qquad (6)$$

where $(1 - exp(-\lambda_i \cdot t))$ is the probability that document i is requested at least once in the period t.

4 Numerical Results

In this section we pick some reasonable values for the different parameters in the model to obtain some quantitative results.

The number of documents N_j in the Web at the beginning of the year j=1998 is about 250 millions [4]. The Web is increasing at a rate such that the number of documents get duplicated every year ($x = 2$) [4]. We consider that the HTTP traffic in the Internet backbone is equal to 1000 document requests per second [3]. This traffic will grow 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 [3].

We consider that there is no relationship between the update period of a document and its popularity [5]. Thus, we assume that all Web documents expire randomly with the same average update period $\Delta_i = \Delta$. We take the percentage of all document requests that are for non-cacheable objects as 10%, i.e. $\theta = 0.1$ [12] [17].

4.1 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 [3] with a baseline of 9 GBytes in 1998 (Figure 3). In a few years it will be easy to find disks with capacities close to TBytes at current prices [9].

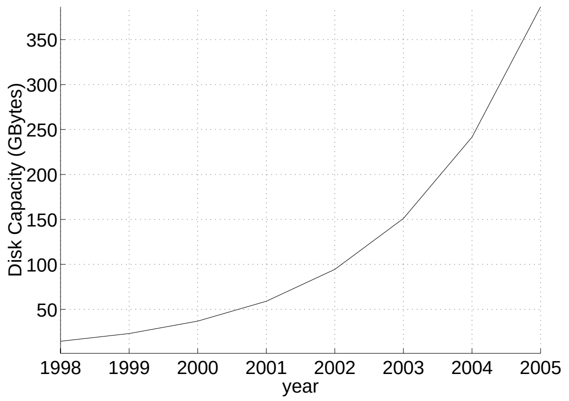

Figure 3: Disk Capacity Trend.

With such large storage capacities it will be feasible to store almost the whole Web on several disks at different points in the network. In figure 4 we show the disk capacity needed to store all the Web documents assuming that any Web document is requested at least once in an update interval ($R_i > 0$). The values presented are an an upper bound for the storage requirements in a cache. Taking an average document size of $S = 100$ Kbytes, and $S = 200$ Kbytes the total number of documents in the Web can be stored in several hundreds of TBytes (figure 4). 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.

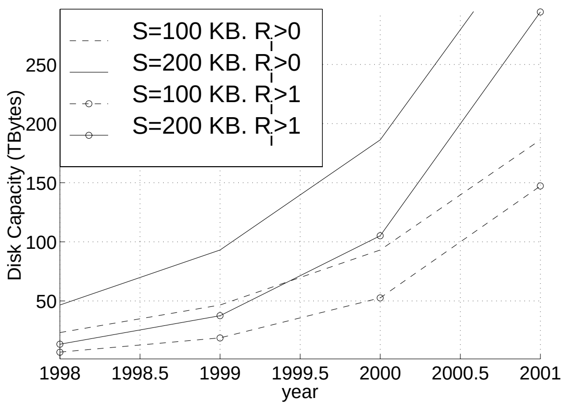

Figure 4: Disk capacity needed to store all Web documents requested at least once and disk capacity needed to store all Web documents requested more than once. $S = 100$ Kbytes, $S = 200$ Kbytes.

Given the long-tail distribution of Web documents, there will 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 thus it is useless to place these kind of documents in all the caches. A satellite distribution can take the decision to broadcast only those documents that are requested more than once ($R_i > 1$) in an update period. Given this situation, the necessary storage capacity drops to 6 Tbytes in year 1998 (Figure 4) and increases at rate equal to rate at which the HTTP Internet backbone traffic increases (x2.8 per year).

Usually individual clients can not afford to buy disks with enormous capacities as it can be the case of an ISPs. When the capacity of the disks may not be enough to store all documents in the Web, clients can perform filtering policies to remove those documents that do not match their preferences and thus decrease their storage requirements (see Section 5).

4.2 Latency Analysis

If documents are hit at network levels close to the clients, the latency to retrieve a document is small. The higher levels of the network (national and international networks) are usually very congested. Local networks carry less traffic and the response times to transmit a document to the clients is small. Given the hierarchical topology of the Internet with very congested top levels and much less congested low levels, we quantify the latency as the number of links needed to hit a document.

Lets give a simple example. Assume that a document expires every day and that receives 1000 requests in this period of time. Lets assume that there are 100 metropolitan caches that issue requests for that document with equal probability. If no caching hierarchy is in place the first client asking for the document in every cache will travel to the origin server to fetch the document. The remaining 9 clients in every cache hit the document locally. If a caching hierarchy is in place the first client out of the 1000 clients fetches the document from the origin server. As the document travels to the client a copy is placed in the intermediate caches in the path from the origin server to the client. Next requests hit the document at lower levels of the caching hierarchy.

If a satellite distribution is in place, the document is pushed to all the caches in any level of the network after the first client fetches the document. The probability that the 999 remaining clients hit the document at closer caches is much higher than in the hierarchical caching scheme. As far as more caches get connected to the satellite distribution, clients travel fewer number of links to hit the document.

To quantify the latency we calculate the average number of links that a client needs to travel to hit a document. We model the underlying topology of the Internet as a full O-ary tree ($O = 4$). The height of the tree, which is the distance between the clients and the origin server, is equal to 10 links. We set the distance between the clients and the metropolitan caches to one link. We also set the distance between the metropolitan caches and the regional caches and between the regional caches and the national cache to 3 links. Client caches are disabled. Due to space limitations we omit a detailed analysis of how to calculate the number of links to hit the document, which can be found in [15].

We will consider three different situations: i) no satellite push is available and all documents are distributed through a caching hierarchy, ii) a caching hierarchy and a satellite push are in place with regional caches connected to the satellite distribution, iii) a caching hierarchy and a satellite push are in place with metropolitan caches connected to the satellite distribution.

Figure 5 shows the expected number of links traversed by the n-th document request arriving in an update period. In all three situations, the first client asking for a document always needs to travel 10 links to hit the document at the origin server. Next clients hit the document at closer levels while the document is up-to-date. If there are many requests for the same document last requests hit the document at the metropolitan cache. If documents

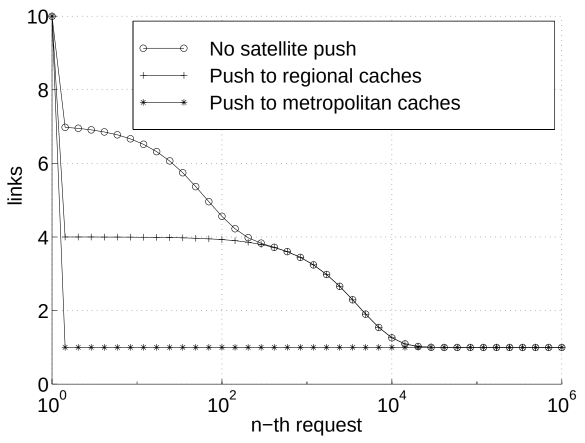

Figure 5: Average number of links traversed before a cache hit occurs for the n-th request

are pushed via the satellite to regional or metropolitan caches, the average number of links that a single client traverses to hit the document is lower than when a satellite distribution is not used. Thus, a satellite distribution to regional or metropolitan caches reduces the latency significantly for the first requests of a document.

4.3 Hit Rate

The higher the number of documents hit in the cache, the higher the bandwidth savings and the lower the latencies. In this section we calculate the hit rate achieved by any cache connected to the satellite distribution as a function of average update period Δ, and we compare it with the hit rate achieved by a metropolitan cache which is not connected to the satellite distribution.

Table 1 shows the hit rate for a cache connected to the satellite distribution and for a metropolitan cache not connected to the satellite distribution. In order to calculate the hit rate for a metropolitan cache that is not connected to the satellite distribution we model the Internet as described in Section 4.2 and assume that document requests are uniformly distributed between all metropolitan caches [15]. Using equation (1) for year $j = 1998$ we get the hit rates for a cache connected to the satellite distribution. From table 1 we observe that caches connected

Δ	1 hour	1 day	10 days
Hit rate Satellite	87 %	88 %	90 %
Hit rate no Satellite	10 %	33 %	53 %

Table 1: Hit rate for a cache connected to the satellite distribution and for a cache not connected to the satellite distribution. 10% of uncacheable objects

to the satellite distribution can greatly increase their hit rates from $10\% - 50\%$ to values close to 90%. The hit rate for caches connected to the satellite distribution does not significantly vary when the update period of the Web documents changes. Additionally we have observed that the hit rate does not significantly change as new documents appear. Thus, due to the high hit rates achieved with a satellite distribution ISPs save a lot of bandwidth and documents can be fetched from lower level caches of the network at lower latencies.

4.4 Bandwidth

As we discussed in Section 3.2, the satellite needs to keep sending all document updates and newly appearing documents in the Web. To calculate the bandwidth needed to broadcast new document updates we need to consider that a Web document is built with several Web objects (images, java applets, text...). When a Web document is updated not all its objects get modified if not only some of them.

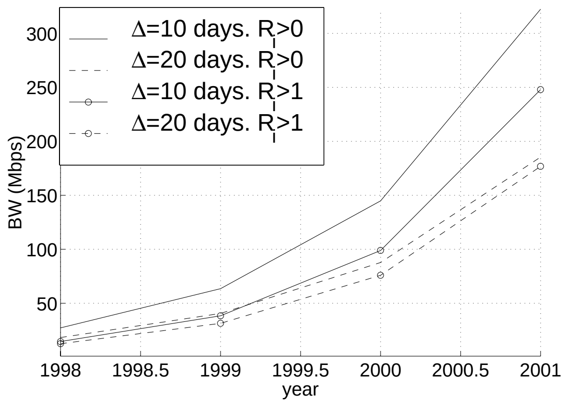

Figure 6: Required bandwidth at the satellite to transmit new documents and document updates: i) all documents requested at least once in an update interval are transmitted across the satellite, ii) only documents requested more than once in an update interval are transmitted across the satellite. 20% of objects in one document are updated

Using equation 4 we plot in Figure 6 the necessary bandwidth at the satellite to keep broadcasting new 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 $S = 100$ Kbytes and consider that only 20% of the document is modified every update period. If 40% of the document is updated every update period, the needed bandwidth gets duplicated. From Figure 6 we can observe that the bandwidth needed to send documents through the satellite is around 30 MBps for year 1998 and that keeps increasing with a rate close to 1.8 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 that Web documents appear in the Web (x2 per year). However, we can observe that even if the number of Web documents gets duplicated every year, the bandwidth requirements increase at a rate of 1.8 the first years. This is due to the fact that many of the appearing documents will be non-popular documents which receive less than one request in an update interval and therefore are rarely transmitted through the satellite. As the HTTP traffic increases, the bandwidth needs to be increased by a factor close to 2 per year because the number of documents that are not requested in an update interval is very small.

As discussed in section 4.1 there is a large group of documents that are only requested one time or less in an update interval. The satellite distribution can take the decision to broadcast only documents that are requested more than once in one update interval and thus reduce the bandwidth used in the satellite. From Figure 6 we can see the bandwidth required when the satellite only transmits documents requested more than once ($R_i > 1$). For an update period of $\Delta = 10$ days, we see that sending only those documents requested more than once saves about 30% of the bandwidth at the satellite. When the update interval of the document increases, there are fewer documents that only receive one request in that interval. Thus, the bandwidth savings decrease when the update interval increases.

It is clear that transmitting only those documents that are requested multiple times in an update interval gives good bandwidth savings. The counter part of sending less documents through the satellite is that the hit rate gets reduced. However, we have quantified the reduction in the hit rate and we have seen that the hit rate is reduced by less than two percent.

A satellite distribution of an analog television channel uses a bandwidth of 27 Mbps. Therefore with the capacity needed to broadcast one TV channel a push-satellite distribution can cope with almost all Web documents in the year 1998. For the master distribution center to send at a rate of 30 Mbps towards the satellite, it needs to receive documents at the same rate from the Internet. Thus, the master needs an access link of at least 30 Mbps.

5 Filtering mechanisms

The filtering mechanisms that we present in this section are aimed at reducing the number of documents that are broadcasted via the satellite or kept in the cache. A good filtering mechanism is such that when some documents are discarded the performance of the system (hit rate, latency to the clients) is not reduced. In previous sections we have already introduced a simple filtering mechanism which avoids sending documents that are requested once or less in an update interval.

Both, the master and the cache need to perform some filtering mechanism. The master distribution center may need to filter some documents due to limitations in the bandwidth at the satellite. The cache may need to filter some documents due to space constraints.

5.1 Cache Side

Even if a cache might have enough space to store all documents in the Web, users of a given cache are not interested in all Web documents. For example, most caches in Europe could safely filter all documents written in an Asian language. The cache can apply a certain filtering policy to remove the documents that do not match the clients preferences and thus free some space for other multimedia applications.

On the other hand, if document sizes start increasing and the cache size becomes limited, the cache needs to decide which documents to store locally. The cache can apply a number of policies to reduce the amount of documents kept without significantly reducing the hit rate.

First, the cache can apply a broad filtering, i.e. remove all documents *.jp but not *.edu. All expired documents could also be purged. Second, the cache can apply a purge routine that purges documents that do not match a long-list of key words. If after this filtering, the cache still needs to purge more documents the following actions can be taken:

- Remove documents from sites that no local user has ever visited
- Remove documents from the bottom of document trees.
- Re-encode images and videos [10]

5.2 Master Side

The master needs to discard some documents when the transmission rate at the satellite is insufficient. First, the master can apply a very simple policy and send only those documents requested more than once. If a hit metering mechanism is in place [13], documents can be shipped to the master with an indication of how popular the document is. Thus, the master can better decide which documents to send. Additionally the master can also re-encode images and videos to reduce the amount of information that is transmitted through the satellite.

Content providers using a satellite distribution will have high bandwidth savings in their Internet access links and lower server's load. Thus, another way for the master to reduce the number of documents distributed via the satellite is by charging content providers. Only the documents from those content providers that are subscribed with the master distribution center will be transmitted to the satellite.

6 Economical Implications and Future Scenarios

Satellite distribution of Web documents has important economic implications for both content providers and ISPs.

The content provider's Web servers will see fewer requests, since many more requests are now serviced by the caches. Thus, content provider's Web server can significantly reduce the rate of its access link connection to the Internet and the capacity of its server. A content provider can provide high-speed access to its site at low cost.

A satellite distribution is filling ISP caches with many documents almost for free. Having such a high number of documents increases the hit rates in the caches. Most of the requests get hit in the ISP cache saving expensive bandwidth in the terrestrial links and avoiding the overhead of communicating with neighboring caches. The master distribution center could also start charging ISPs connected to the satellite distribution because ISPs are increasing considerably their hit rates and saving bandwidth.

If clients with local caches are also connected to the satellite distribution they do not suffer the last mile problem for the Web traffic. Web documents come through the satellite and can be browsed locally. The network is rarely used by the clients. The network is used only to check document's consistency or to access non Web documents. Due to the high number of documents kept locally by the client local search engines should be developed from to benefit from the possibility to hit documents locally [1] [2].

Transmitting documents through the satellite to local caches is a distribution scheme that will become even more relevant when the Web and the TV get integrated. The Television is currently the most important mass media. The Web is growing exponentially and becoming another important mass media. The arrival of the digital television will trigger the development of television and Web integration. A satellite distribution can be used to push Web documents related with a certain TV program into clients caches and into a caching hierarchy [14]. As the pushed documents are very related with the actual TV program (i.e. additional weather maps on the weather report, information about the players in a football match..), the probability that clients request the proposed pushed-documents is higher. Additionally clients know that the pushed documents can be browsed locally with minimum delay and without using the Internet, which makes these documents even more attractive. The pushed documents come as a value-added service to the TV information.

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 found in a cache are requests for documents that no one has fetched before in that cache and for uncacheable objects. Uncacheable objects may become cacheable if some intelligence is place in the caches to cope with dynamic content (rotating banners for advertisements, cookies, etc) [7]. To reduce the number of requests for documents that no one has fetch before a high community of clients needs to be shared.

A satellite distribution of Web documents into large caches can reduce the number of requests that are first requests for a certain document almost to zero. The satellite fills the caches with all documents requested by a large community of clients. Documents get into the caches almost for free and significantly alleviate the congested Internet links. In this paper we have presented analytical models and quantitative results showing that a satellite distribution of Web documents into caches is feasible. The satellite distribution would need a capacity equivalent to the bandwidth of a TV satellite channel to transmit all appearing Web documents and document updates during year 1998. The hit rates in the caches get close to 90 %.

When the satellite capacity or the cache storage capacity is limited some documents must be discarded. We have presented several filtering mechanisms to cope with the lack of bandwidth in the satellite and the storage limitations in the caches. Additionally we have discussed the economical implications of a satellite distribution and presented some future scenarios. We are currently performing trace-driven simulations to better confirm the analytical results presented in this paper.

8 Acknowledgments

The support of P. Rodriguez by the European Commission in form of a TMR (Training and Mobility for Researchers in Europe) fellowship (ERBFMBICT97263 9) is gratefully acknowledged. We would like to thank Keith W. Ross for his help and the fruitful discussions. Eurecom's research is partially supported by its industrial partners: Ascom, Cegetel, France Telecom, Hitachi, IBM France, Motorola, Swisscom, Texas Instruments, and Thomson CSF.

References

[1] "The Microsoft White Paper about Webcasting IE 4.0".

[2] "The Netscape Netcaster.".

[3] 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.

[4] 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.

[5] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "On the Implications of Zipf's Law for Web Caching", In *3rd International WWW Caching Workshop*, June 1998.

[6] Broadcast Satellite Services, "http://www.isp-sat.com".

[7] P. Cao, J. Zhang, and K. Beach, "Active Cache: Caching Dynamic Contents on the Web", , University of Wisconsin-Madison, 1998.

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

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

[10] J. Kangasharju, Y. G. Kwon, and A. Ortega, "Design and Implementation of a Soft Caching Proxy", In *3rd International WWW Caching Workshop*, Manchester, UK, June 1998.

[11] C. Liu and P. Cao, "Maintaining Strong Cache Consistency in the World-Wide Web", In *Proceedings of ICDCS*, may 1997.

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

[13] J. Mogul and P. J. Leach, "Hit-Metering and Usage-Limiting for HTTP", Internet Draft: draft-ietf-http-hit-metering-02. Work in Progress., Internet Engineering Task Force, October 1997.

[14] P. Rodriguez, J. Gafsi, and J. Nonnenmacher, "A more Attractive and Interactive TV", In *W3C Workshop on Television and the Web*, Sophia Antipolis, France, July 1998.

[15] P. Rodriguez, K. W.Ross, and E. W.Biersack, "Distributing Frequently-Changing Documents in the Web: Multicasting or Hierarchical Caching", *To appear in Computer Networks and ISDN systems*, 1998.

[16] SkyCache, "http://www.skycache.com".

[17] R. Tewari, M. Dahlin, H. M. Vin, and J. S. Kay, "Beyond Hierarchies: Design Considerations for Distributed Caching on the Internet", , The University of Texas at Austin, Feb 1998.

[18] D. Wessels, "Squid Internet Object Cache: http://www.nlanr.net/Squid/", 1996.

[19] 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.

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