Improving the WWW: Caching or Multicast?

Pablo Rodriguez   Ernst W. Biersack   Keith W. Ross

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

March 30, 1998

Abstract

Latency is crucial for the success of the WWW. Access to popular pages can lead to high latency due to overloaded servers. Caching is one solution, multicast is another. In this paper we compare the delivery of popular and changing Web documents through a caching hierarchy with the delivery of the same documents through a multicast distribution. We have found that except for popular documents that change very fast, caching is preferable to multicast. However, for hot and short-lived documents a multicast distribution reduces the latency to the receivers, saves network bandwidth, and reduces the load on the original server. Therefore, a distribution scheme for Web documents on the Internet should implement both solutions, caching and multicast.

Keywords: Multicast, Caching

1 Introduction

The exponential growth of the Internet and the World Wide Web is causing a great demand for bandwidth and server capacity. As a result the latency that receivers perceive is increasing. Scaling the Internet by simply adding more resources (bandwidth and processing power) may not suffice and therefore incentive schemes such as differential pricing may be necessary. On the other hand, there are other mechanisms aimed at using available resources (server, bandwidth) more efficiently and reduce the latency to the receivers: **Caching and Multicast**.

Caching and multicast are two different distribution methods. We find that each of these distribution methods is best suited for the delivery of different Web documents. As a result, we claim that both, caching and multicast should be used together. For popular documents that do not change very often, a caching hierarchy is the best way to reduce the latency to the receivers. Only the first request experiences a high delay since the document must be fetched from the original server. Further requests find the document at local caches which are connected to the receivers via high capacity networks.

In addition to popular documents that do not change very often, there also exists a large class of documents such as news, stock market data, or auction data that, besides being of interest to a large number of receivers, change frequently. We argue that these documents can be best delivered using a **continuous multicast push (CMP)** [20].

In this paper, we compare the latency perceived by a receiver delivering a document through a caching hierarchy with that through a multicast distribution. We find that except for the case of highly popular, rapidly changing documents, caching is preferable to CMP. For very popular and fast changing documents, the time to obtain the first packet of the document is higher on a caching distribution than on a multicast distribution (Section 4.1). Additionally, in the case of popula and fast-changing Web documents, the upper caches get overloaded with many requests. This increases the completion time to obtain the whole document on a caching hierarchy compared to that on a CMP distribution (Section 4.2).

Based on these results we view CMP as one of the three complementary delivery options integrated on the World Wide Web: *Caching, AMP, and CMP*:

- **Caching**: Popular documents that do not rapidly change and documents that are rarely requested are distributed via a caching hierarchy.
- **Asynchronous Multicast Push (AMP)** [6] [16]: This method can also be used for popular documents that do not frequently change. Requests for the same document are accumulated over a time interval, and answered together via multicast. The drawback of this method is that the grouping time increases the latency to the receivers. However, AMP saves a lot of bandwidth on the network. If receivers can tolerate a certain delay (i.e. distribution of software during the nights) then, they could benefit from lower tarifications due to the savings obtained on the network.
- **Continuous Multicast Push (CMP)**: A document is continuously multicasted on the same multicast address. This delivery method is used for very popular documents that change very frequently. A Web server using CMP continuously multicasts the latest version of a popular Web document on a multicast address (every document is assigned a different multicast address). Receivers tune into the multicast group for the time required to reliably receive the document and then leave the group.

  The CMP distribution works as follows:

  1. A Web server monitors the number of requests for a document to decide which documents to multicast. Only popular and frequently changing documents are sent via CMP.
  2. The server takes the popular document and sends it cyclicly in a multicast address.
  3. Receivers obtain a mapping of the document's name (URL) into a multicast address and then join the multicast group. Receivers stay in the multicast group until they have received the Web document reliably.
  4. The server keeps monitoring the number of requests and stops multicasting the document if there are no more receivers.

2 Caching vs Continuous Multicast Push

2.1 Caching

Caching, reduces the bandwidth usage and latency to the receivers on the Internet. Caching takes place at the application layer and allows for an incremental deployment of caches. Caching is already a fact of life in much of the Internet [2]. Most **ISPs (Internet Service Providers)** and organizations connected to the Internet have been installing caches to reduce the bandwidth and decrease the latency to their users [2] [5] [4] [18] [13]. However, caching does not come for free and there are still open issues relating to it.

- Installing a cache requires additional resources such as computers, disks, software, etc.
- Caches need to cooperate together to increase the hit rate [5] [22] [21]. This creates additional overhead.

- Caches need to maintain document consistency and provide the user with the most recent update.

In fact, the effort of installing a caching hierarchy resembles the effort that was required to put in place the first experimental multicast overlay network called **MBONE** [10] [14]. A cache hierarchy is very similar to a multicast distribution scheme but with "application hop"-by-"application hop" congestion control and reliability. Documents are stored at the different caching levels based on the principle that some other receivers may be interested on the same document at a later time assuming that the document is still up-to-date.

When a cache receives a document the first time, it can start forwarding the document to the lower cache levels without waiting for the reception of the whole document. Therefore, there is no such "store-and-forward" delay on a caching hierarchy as there may be on a file system. However, going through a caching hierarchy to obtain a document has several additional delays that a multicast distribution has not:

- *Resolution delay*: This delay accounts for the time to check if the document is kept by any cache at that level (ICP queries [26], hashing function [21], routing [24]...)
- *TCP delay*: This delay is due to the slow start phase of the different TCP connections between every cache level [17]. The slow start is more relevant when the completion time of the document is small. The effect of this delay gets very reduced when persistent TCP connections are allowed [11].
- *Queuing delay*: This delay is due to queues on busy caches.
- *Server delay*: This delay is due to busy servers that need to deal with many requests for document updates from several root caches.

The Harvest cache [5] and its public domain descendant Squid [25] are becoming the most popular caching hierarchies on the Internet [2]. In this paper we compare a caching hierarchy with a multicast distribution. Perhaps, more efficient caching schemes can be implemented that solve the limitations of a caching hierarchy [23]. However, any caching scheme implemented will follow the same principle of storing documents and deliver them to later receivers from closer caches. For popular and fast changing documents it doesn't seem to be a very good idea to store documents that expire instantaneously.

Some recent studies confirm the fact that even for infinite size caches the achieved hit rate in a caching hierarchy is limited [23] [2]. A caching hierarchy is not able to satisfy all requests due to different kind of misses:

- *First-Access*: Misses occurred when accessing documents the first time.
- *Capacity*: Misses occurred when accessing documents previously requested but discarded from the cache to make space for other documents.
- *Updates*: Misses occurred when accessing documents previously requested but already expired.
- *Uncacheable*: Misses occurred when accessing documents that need to be delivered from the final server.

First-access misses are much higher than any other kind of misses [23] and may account for the 20% of all requests. We expect capacity misses to be a secondary issue for large-scale cache architectures because it is becoming very popular to build shared caches with small number of capacity misses. Therefore, we assume an infinity cache space. For the rest of the paper we do not consider Capacity or Uncacheable misses and only consider Updates and First-Access misses.

Given that a cache keeps all previously requested documents:

- The first request for a document travels all the caching hierarchy until the original server accounting for one First-Access miss.
- Later requests for the same document are delivered from the institutional cache (assuming that receivers' local caches are disabled). When the document expires a new request needs to travel again to the original server accounting for an Update miss.

2.2 CMP

CMP takes place at the transport layer (of the OSI model) with reliability and congestion control ensured by the end systems (server and clients). In the context of the Internet, CMP requires that the network connecting a server with its clients is multicast capable: a single packet sent by a server will be forwarded along the multicast tree (see Fig 1).

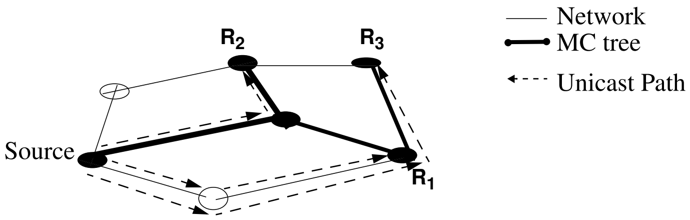


Figure 1: Network topology with multicast tree.

Where the multicast tree forks off, the multicast router will replicate the packets and send a copy on every outgoing branch of the multicast tree. Multicast routers on the Internet were first introduced via an overlay network called MBONE consisting of computers that executed the multicast routing software and that were connected via tunnels. While today the multicast routing software is part of any new router that is installed, not all the existing routers have been enabled to be multicast capable. Therefore, multicast routing on the Internet is not yet everywhere available.

A multicast distribution using CMP does not suffer problems of over-loaded servers or caches. The multicast server does not deal directly with the receivers reducing the server complexity and scaling very well. Receivers obtain at any moment the last available update without incurring on the overhead of checking for the updated document on all the cache levels.

On the other hand, a multicast distribution uses bandwidth efficiently by sharing *all* common paths between the source and the receivers. In the current caching hierarchy, communication is generally done via unicast between the different cache levels (there have been a number of proposals to communicate caches via multicast [26] [15]).

A continuous multicast push for popular documents that change frequently seems to be the best way to deliver this kind of information. However, multicast distribution of Web documents on the Internet is still in its infancy as a viable service; in fact, very few network providers offer it as a service [12]. A continuous multicast distribution also requires some additional mechanisms:

- Session servers or a similar mechanism are needed to map the document's name into a multicast address.
- A Web server needs to monitor the number of document requests and their rate of change to decide which documents to multicast and when to stop multicasting them.
- There is an overhead in the multicast capable routers to maintain state information for each active multicast group.
- There is also an overhead due to the join and prune messages needed for the multicast tree to grow and shrink depending on the location of the receivers.
- Multicast congestion control is still an open issue.

We claim that due to the varying nature of the different Web documents, there is a room for both caching and continuous multicast distribution.

3 Model

We model the network connecting the server and the receivers as a succession of network clouds at different administrative levels (national, regional, and institutional) (Figure 2). The national network joins the different regional networks inside one country. A regional network joins all institutional networks in one region.

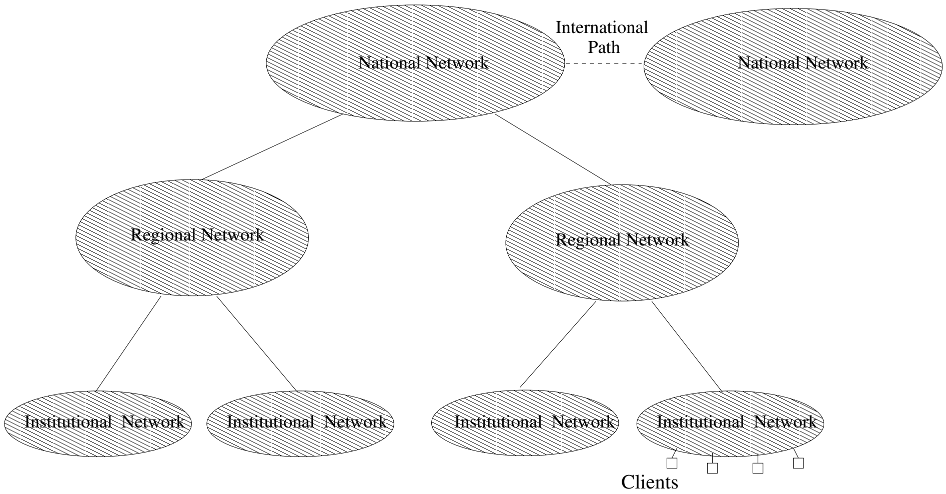


Figure 2: Network topology

In order to have a common basis for the comparison of caching versus multicast we model the underlying network topology as a full O-ary tree (Figure 3)

Multicast The nodes of the O-ary tree are multicast routers. The **original server** is connected to the receivers via a multicast tree. In the following we assume the original server to be connected to the receivers via a core based tree [3, 7]. The server sends to the core, which is the root for a shortest path tree [8], where a receiver is connected to the core via a shortest path through the network.

Caching Caches are usually placed at the access points between two different networks to decrease the expenses of traveling through a new network. For caching a 3-level cache hierarchy is modeled, consisting of national, regional, and institutional caches. In one country there is one national network with one national cache. There are O^H regional networks and every one has a regional cache. There are O^{2H} local networks and every one has an institutional cache. An institutional cache is connected to several receivers. Caches are placed on height 1 of the tree (level 1 in the cache hierarchy), height $H+1$ of the tree (level 2 in the cache hierarchy) and height $2H+1$ of the tree (level 3 of the hierarchy). If a requested document is not found in the cache hierarchy the national cache requests the document directly from the server.

Receivers are only on the leaves of the tree and not on the intermediate nodes. Receivers' local caches are disabled.

Parameter Setting

- z: Number of links on the **International Path**. The International Path models the international link joining the final server with the national cache or the final server with the core of the multicast tree.

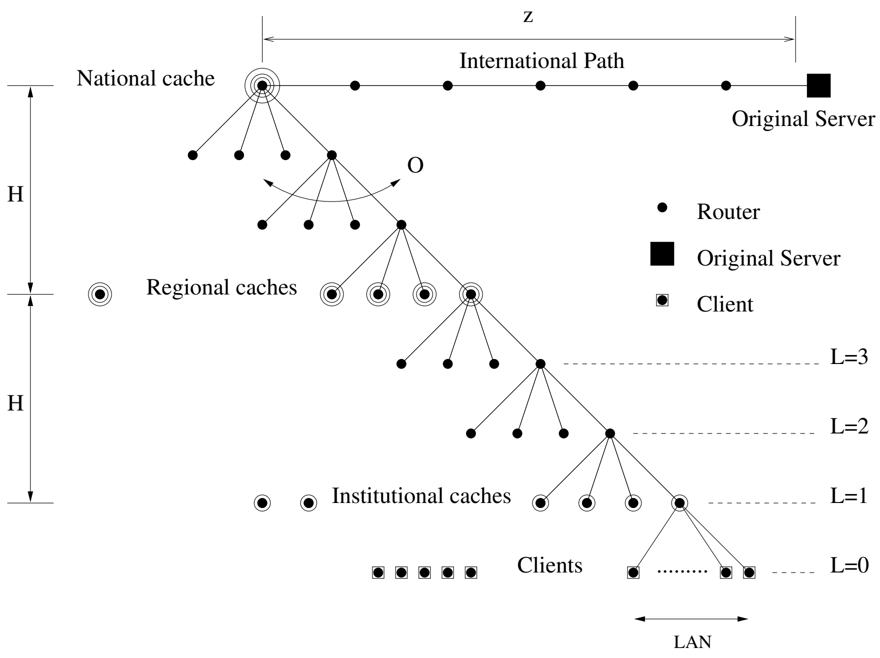


Figure 3: The tree model for Multicast and caching.

- S: the Web document size in kBytes.
- T: the period of change of the document in seconds.
- λ_{LAN}: the average aggregated request rate from all receivers in one LAN for the same document. The inter-arrival time between requests τ is exponentially distributed (the number of arrivals X is a Poisson distribution). The cumulative distribution function of τ is given by:

$$\begin{aligned} F(\tau, \lambda_l) &= 1 - exp(-\lambda_l \cdot \tau) \\ \lambda_l &= \lambda_{LAN} \cdot O^{l-1} \qquad ,1 \leq l \leq 2H+1 \end{aligned}$$

  Going up the tree in Figure 3 requests from several LANs merge at the next level. Merging m Poisson processes with request rate λ each, results again in a Poisson process with arrival rate $m\lambda$. Given the tree model in figure 3 the request rate λ_l refers to the aggregated request rate from all LANs below the multicast tree rooted at level $L = l$.
- $\lambda_{tot} = \lambda_{LAN} O^{2H}$: The aggregated number of requests from all LANs for the same document.
- $\lambda^{H,F}_{LAN}$: average aggregate request rate from one LAN for *one* Hot-Changing document.
- $\lambda^{H,F}_{tot} = \lambda^{H,F}_{LAN} * O^{2H}$: average aggregate request rate from all LANs for *one* Hot-Changing document.
- β_{LAN}: The average aggregated request rate from all receivers in one LAN for *all* documents.

We classify all Web documents requested on one LAN as follows:

- *Hot-Changing*: Documents that are very popular and change every period T. They account for $\beta^{H,F}_{LAN}$ req/sec. There are $N_{H,F}$ Hot-Changing documents.

$$\beta^{H,F}_{LAN} = \lambda^{H,F}_{LAN} \cdot N_{H,F}$$

- *Hot-Stable*: Documents that are very popular but never change. They account for $\beta_{LAN}^{H,S}$ req/sec.
- *Cold*: Documents that are only requested once on a LAN. They are responsible for the First-Access misses. They account for β_{LAN}^{C} req/sec.

$$\beta_{LAN} = \beta_{LAN}^{H,F} + \beta_{LAN}^{H,S} + \beta_{LAN}^{C}$$

Recent studies [9] [23] have analyzed the percentage of requests that every document type accounts for. Some indicative values are: $\beta_{LAN}^{H,S} = 0.5\beta_{LAN}$, $\beta_{LAN}^{H,F} = 0.15\beta_{LAN}$, $\beta_{LAN}^{C} = 0.35\beta_{LAN}$.

- $Hit(l)$: Hit rate for all documents at level $L = l$ [23] [1]. $Hit(1) = 0.5$, $Hit(H+1) = 0.6$, $Hit(2H+1) = 0.7$.
- $Hit_c(l)$: Hit rate for Cold documents at level $L = l$. $Hit_c(1) = 0$, $Hit_c(H+1) = 0.28$, $Hit_c(2H+1) = 0.57$

A country is modeled by the choice of the tree parameters as follows:

- $O = 4$ as nodal outdegree of the MC tree.
- $H = 3$ as the distance between cache hierarchy levels, yielding $O^H = 64$ regional caches and $O^{2H} = 4096$ institutional caches.

The document changes periodically every T seconds and the document is delivered via an homogeneous network, either by multicast from the server, or by unicast from a 3-level caching hierarchy.

4 Latency

The **Total Latency time** T_{tot} can be divided in two parts:

- **First-Packet Time** T_f: The time between one receiver makes a request and the time the first packet arrives at that receiver.
- **Completion Time** T_c: The time between the arrival of the first packet and the time that the receiver completes the reception of the most up-to-date document version.

$$T_{tot} = T_f + T_c$$

4.1 First-Packet Time T_f

4.1.1 Multicast

The first-packet time is measured up to the point in time where the receiver gets the first bit of the document. The following random variables are defined:

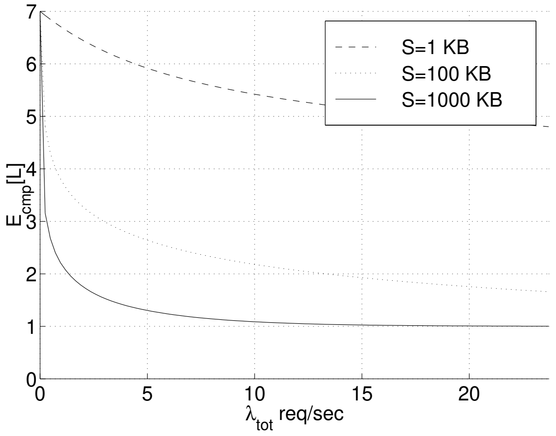
- L: The number of links a new request has to travel on the multicast tree or on the caching hierarchy to meet the document.
- μ_{cmp}: The multicast transmission rate seen by a receiver. This rate fluctuates along the day depending on the network congestion experienced at the different hours. We choose some average representative values.
- d: The propagation and transmission delay on one link, homogeneous for all links. This delay does not account for the transmission time of the whole document which is treated on section 4.2.

The expected first-packet time for a multicast and a caching distribution is:

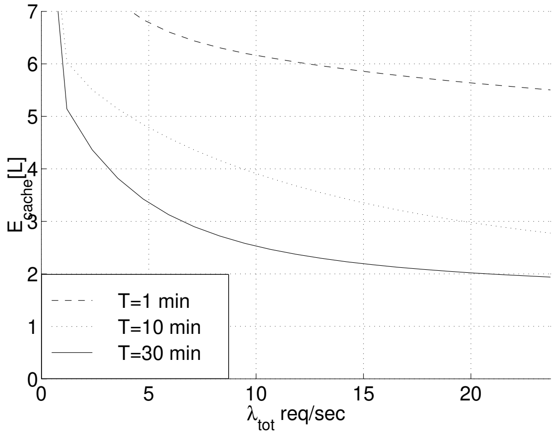
$$E_{cmp}[T_f] = 2d \cdot E_{cmp}[L]$$
$$E_{cache}[T_f] = 2d \cdot E_{cache}[L]$$

On a caching hierarchy a TCP connection is openned between every caching level before starting the transmission of the Web document. On a multicast distribution the delivery is open-loop and there is no previous connection between the receivers and the source. Therefore, when we calculate $E_{cache}[L]$ we need to scale it by a factor 1.5 due to the three-way handshake TCP protocol.

We have analyzed the average number of links traversed by a request on a multicast tree $E_{cmp}[L]$ and on a caching hierarchy $E_{cache}[L]$ depending on the total number of requests for a document, the document size and its rate of change [19]. In this paper we do not present the detailed analysis of the completion time but we show some of the results obtained.



(a) Multicast



(b) Caching

Figure 4: Average number of traversed links $E_{cmp}[L]$, $E_{cache}[L]$ for different update periods and document sizes depending on the total request rate. $\mu_{cmp} = 1$ KB/sec.

Figure 4(a) shows $E_{cmp}[L]$ for different document sizes depending on the total request rate λ_{tot}. The height of the tree is $2H + 1 = 7$. When the number of requests is very small, it is very probable that a join has to travel all the way to the final server in order to meet the multicast tree. A new request can not share any branch of the multicast tree built for past requests because it is already shrunk. However, if the number of requests is high a new request will meet the multicast tree at a lower level.

For very small documents, the tree shrinks very fast reducing the probability of meeting the tree at a low level. However, for big documents the probability of meeting the tree at a low level increases reducing the average number of links traversed.

Figure 4(b) shows $E_{cache}[L]$ depending on λ_{tot} for different update periods T. We see that if the document is rarely requested, the average number of travelled links needed to meet a cache with an up-to-date document increases. For high request rates, a new arriving request meets the up-to-date document at a closer caching level. However, even if the request rate continuous increasing, there is limit on the minimum number of links traversed. This is due to the fact that every period T the document expires and needs to be updated directly from the final server. If the document changes very fast, the number of links traversed clearly increases.

Comparing Figures 4(a) and 4(b) we observe that the number of links traversed to find the most up-to-date document version on a multicast distribution is smaller than on a caching distribution, specially when the document is very popular and changes fast. Even if the document changes only every 30 minutes the number of links traversed on a multicast distribution is smaller than on a caching distribution.

However, the first-packet time T_f is only relevant for the total latency time T_{tot} if the document size is very small. For document sizes $S = 100$ KB we have some results showing that the first-packet time is one order of magnitude lower than the completion time and therefore has small influence on the total latency time.

4.2 Completion Time T_c

The completion time T_c accounts for the time from the first-packet arrival until the time that the receiver completes the reception of the document. We have taken some indicative values for the network capacities at the different levels (Figure 5). We assume that there is no other kind of traffic more than Web traffic going through these networks.

On the upper levels of the hierarchy the links are very utilized to deliver many Web documents while the lower levels of the hierarchy are much less utilized. For example, the International network capacity is shared between many document transmissions reducing the available capacity for one document transmission. On the other hand, on a LAN the congestion problems are much lower and therefore the available capacity for one document transmission is very high.

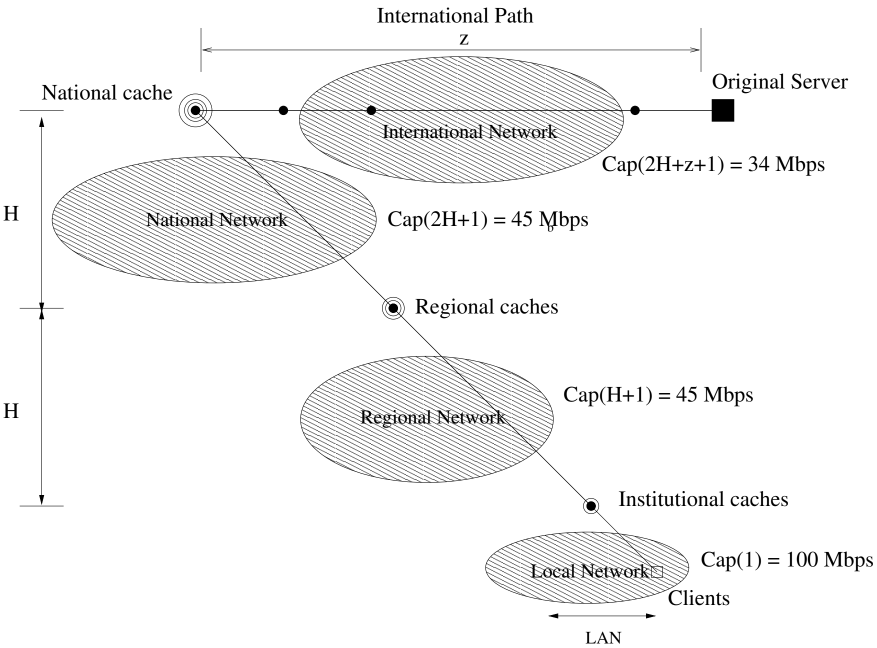


Figure 5: Network and Caches topology

On the previous Section 4.1, we have considered the case of one single Web document with different request and changing rates to calculate the First-Packet time T_f. For the completion time T_c, we need to consider different types of Web documents which share the available capacity on the network (Section 3).

We analyze three different scenarios:

1. *Pure Multicast*: There is no caching hierarchy. The Hot-Changing documents are delivered via CMP. The rest of the documents are delivered directly from the original server sharing the available bandwidth with the Hot-Changing documents.
2. *CMP-Cache*: There is a caching hierarchy where the Hot-Stable documents are stored without any disk space constraints. The Hot-Changing documents are delivered via CMP. The Hot-Stable documents are delivered to the receivers from the institutional caches. The Cold documents can not be delivered from the institutional caches but there is a certain probability that a request for a Cold document is hit at higher cache level.
3. *Pure Cache*: All documents are delivered via the caching hierarchy.

We assume the worst case for a multicast distribution:

- there is at least one interested receiver for all $N_{H,F}$ Hot-Changing document at every moment on all LANs, even if the document does not change every new transmission.

In this situation, a multicast distribution needs to continuously sent the $N_{H,F}$ documents from the original source to all LANs while on a caching distribution the $N_{H,F}$ documents need to be delivered only once every period T and then they are forwarded locally. If receivers were time-synchronised a multicast distribution would also need only one document transmission every period T.

For the three scenarios we calculate the average completion time $E[T_c]$ for a Hot-Changing document and assume that the original servers are placed beyond the International Path. We consider that there are $N_{H,F} = 100$ Hot-Changing documents.

4.2.1 Pure Multicast

Multicasting is an *"end-to-end"* distribution. A document travels continuously all the way from the final server to the receivers. A multicast distribution is limited by the minimum availiable capacity for a document transmission on the path from the source to the receivers.

Given that there is no caching hierarchy dealing with some of the requests, the network capacity at all levels is being shared by all document requests from O^{2H} LANs. The link with more traffic and the lowest capacity is the International path $Cap(2H + z + 1)$.

Therefore, the completion time for one Hot-Changing document is given by:

$$E_{cmp}[T_c] = \frac{S \cdot N_{H,F}}{Cap(2H + z + 1) - (\beta_{LAN}^{H,S} + \beta_{LAN}^{C})O^{2H}S}$$

Where $(\beta_{LAN}^{H,S} + \beta_{LAN}^{C}) \cdot O^{2H}S$ is the capacity needed to answer the Hot-Stable and Cold requests from all LANs.

Figure 6 shows the completion time for a pure multicast distribution depending on the total number of requests for a Hot-Changing document. When the total request rate in a LAN β_{LAN} increases, the number of requests for Cold and Hot-Stable documents increases in a proportional way, therefore reducing the available bandwidth for Hot-Changing documents. When the demand rate exceeds the available capacity, the completion time grows towards infinity.

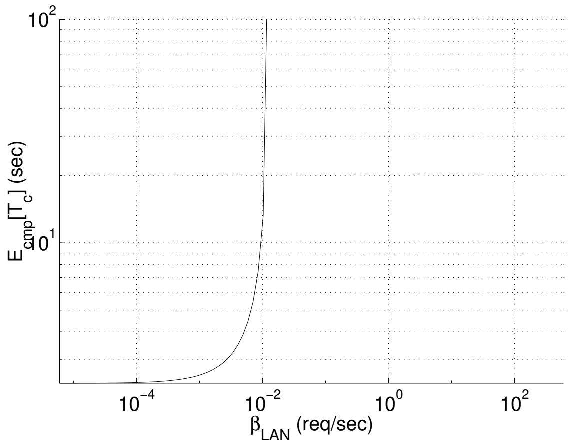


Figure 6: Pure Multicast completion time for a Hot-Changing document for different β_{LAN}. $N_{H,F} = 100$. $S = 100$ KB. $Cap(2H + z + 1) = 34$ Mbps

4.2.2 CMP-Cache

Hot-Changing documents are continuously multicasted from the original server. The caching hierarchy answers all requests for Hot-Stable documents from the institutional caches. Cold documents can be hit on upper caches. However, requests for Hot-Changing documents and Cold documents that were not hit on the caching hierarchy travel all the way to the original server.

The completion time $E_{hybrid}[T_c]$ for a Hot-Changing document that is multicasted from the original server to the receivers is given by the maximum completion time at any hierarchy level $E^l_{hybrid}[T_c]$.

- $E^l_{hybrid}[T_c]$: Completion time to transmit one Hot-Changing document from level l to level $l - 1$.
- $E_{hybrid}[T_c] = max_{l \in \{1,H+1,2H+1,2H+z+1\}} \{E^l_{hybrid}[T_c]\}$

$$E^l_{hybrid}[T_c] = \begin{cases} \dfrac{N_{h,c} \cdot S}{Cap(1) - (\beta^C_{LAN} + \beta^{H,S}_{LAN}) \cdot S} & ,l = 1 \\[2ex] \dfrac{N_{h,c} \cdot S}{Cap(H+1) - \beta^C_{LAN} O^H (1 - Hit_c(1)) \cdot S} & ,l = H + 1 \\[2ex] \dfrac{N_{h,c} \cdot S}{Cap(2H+1) - \beta^C_{LAN} O^{2H} (1 - Hit_c(H+1)) \cdot S} & ,l = 2H + 1 \\[2ex] \dfrac{N_{h,c} \cdot S}{Cap(2H+z+1) - \beta^C_{LAN} O^{2H} (1 - Hit_c(2H+1)) \cdot S} & ,l = 2H + z + 1 \end{cases}$$

Figure 7 shows the completion time at the regional, national and international networks. The maximum completion time for all levels is plot with a continuous line.

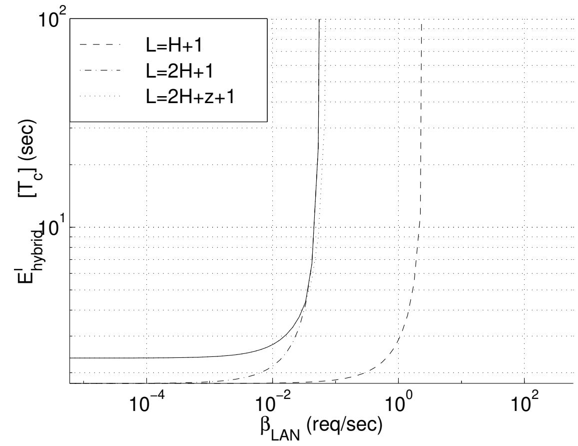


Figure 7: CMP-Cache completion time at the different hierarchy levels for Hot-Changing documents for different β_{LAN}. The continuous line represents the maximum completion time at all levels. $N_{H,F} = 100$. $S = 100$ KB. $Cap(H+1) = 45$ Mbps, $Cap(H+1) = 45$ Mbps, $Cap(2H+z+1) = 34$ Mbps.

4.2.3 Pure Cache

All documents are delivered through the caching hierarchy. A Hot-Changing document travels once every period T from the final server to the institutional caches and then requests are fulfilled directly from the institutional caches.

Therefore, the average completion time is given by:

$$E_{cache}[T_c] = \quad E^1_{cache}[T_c] \cdot (1-\gamma) + max_{l\in\{1,H+1,2H+1,2H+z+1\}}\{E^l_{cache}[T_c]\} \cdot \gamma$$

- γ: Percentage of requests for a Hot-Changing document that see a document update. If every document transmission is a new document update $\gamma = 1$ and we say that the document changes *continuously*. Recent studies showed that 13% of all requests for popular documents see a document update [9].

$$\gamma = \frac{1}{\lambda^{H,F}_{LAN} T} \quad ,0 < \gamma \leq 1$$

In order to calculate E^l_{cache} we proceed in a similar way as in previous sections. We calculate the average request rate arriving to a cache at level l for all documents $\beta(l)$. This request rate is filtered by the hits for Hot-Changing and Cold documents at lower level caches. Then, we use a M/D/1 queue to model the queuing delays on the institutional, regional and national caches as the one on Figure 8.

When $\gamma = 1$ we consider that the completion time on the International network ($L = 2H + z + 1$) for a caching distribution $E_{cache}[T_c]$ is equal to the completion time for a CMP-caching distribution $E_{hybrid}[T_c]$. However, when the document does not expire very often, γ is small and the number of document transmissions sent through the International network is smaller. This increases the available capacity for Hot-Changing documents and reduces the completion time.

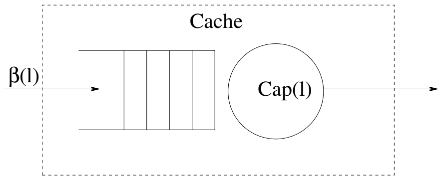


Figure 8: Queuing model for the load on the caches.

As a result, on figure 9 shows $E^l_{cache}[T_c]$ for the international, national and regional networks when the document changes continuously ($\gamma = 1$). The continuous line represents the maximum completion time at all levels. From the plot we observe that most of the time the maximum completion time $E_{cache}[T_c]$ is given by the completion time at the international path. However, when the number of requests is high, the queuing delays on the national cache start being significant increasing the completion time at the national network over the completion time on the international network.

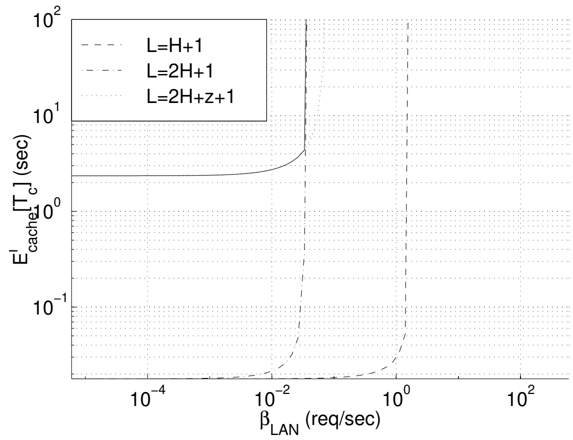


Figure 9: Pure Cache completion time for a Hot-Changing document at the different hierarchy levels for different β_{LAN}. The document changes continuously $\gamma = 1$. The continuous line represents the maximum completion time at all levels. $N_{H,F} = 100$. $S = 100$ KB. $Cap(H + 1) = 45$ Mbps, $Cap(H + 1) = 45$ Mbps, $Cap(2H + z + 1) = 34$ Mbps.

On figure 10 we have plot together the completion time for a Hot-Changing document for the three different scenarios: Pure Multicast, CMP+Cache, and Pure Cache. For the Pure Cache scheme we have varied the percentage γ of requests that see a document update. Changing γ does not affect the completion time of a Pure Multicast or a CMP-Cache distribution because the Hot-Changing documents are being continuously pushed regardless of their changing rate. However, for a Pure Cache distribution when the changing rate is small there are fewer document transmissions on the International network than if the document changes faster, increasing the available capacity for one Hot-Changing document transmission.

We find that a Pure Multicast distribution has the worst completion time of the three different scenarios. The completion time of a Pure Cache distribution is equal or higher than the completion time of a CMP+Cache distribution when every document request sees a new document update ($\gamma = 1$). In most cases the completion time of a Pure Cache and a CMP+Cache are equal. However, when the request

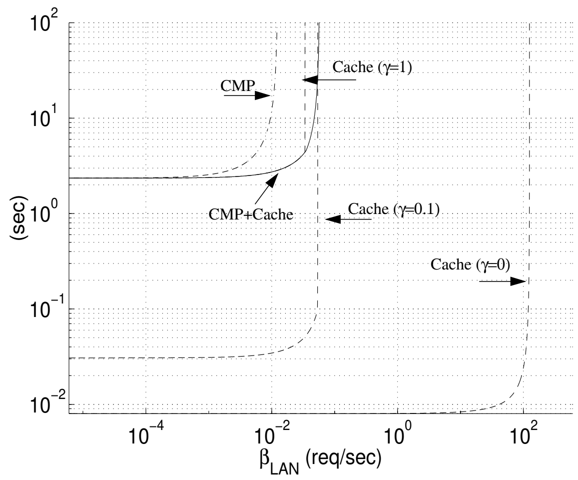


Figure 10: Completion time for a Hot-Changing document on a Pure Multicast, CMP+Cache, and Pure Cache scheme for different values of γ. $N_{H,F} = 100$. $S = 100$ KB. $Cap(H+1) = 45$ Mbps, $Cap(H+1) = 45$ Mbps, $Cap(2H + z + 1) = 34$ Mbps.

rate increases over a certain value the queuing delays on the national cache become important and make the completion time for the Pure Cache distribution increase over the completion time of a CMP+Cache distribution.

When 10% of all requests see a document update ($\gamma = 0.1$) the 90% remaining requests benefit from very low completion times because the document is hit at the institutional cache. Therefore, in most cases a Pure Cache distribution has a lower completion time than a CMP+Cache distribution. However, for high request rates, 10% of the requests experience high queuing delays on the national cache increasing the completion time on a Pure Cache distribution to values similar to the ones of a CMP-Cache distribution.

When the Hot-Changing documents do not change ($\gamma = 0$), all requests except those ones for Cold documents are fulfilled from the institutional caches reducing considerably the completion time.

5 Caching and Multicast: Push Caching

On this section we propose a mechanism combining caching and multicast to reduce the latency to the receivers.

There are several solutions to improve the completion time on a caching hierarchy:

1. Increase the bandwidth of the channel. Increasing the bandwidth the problem disappears. However, there can always be new applications with bigger documents that consume again the available bandwidth. Sometimes the problem is not the bandwidth but the cache itself. On this case more machines are needed to cooperate sharing the load to reduce the latency.
2. Reduce the request rate at every cache by distributing the documents over many caches. This changes the topology model from a hierarchical topology to a *flat* topology [23].
3. Use bandwidth more efficiently. Multicasting updates for popular documents from one cache level to the other saves bandwidth.

A multicast distribution can be used by an up-level cache to send document updates to all lower level cache. In order to exploit the benefits of a multicast distribution all caches at the lower level need to be synchronized on time. If requests for updates are received at the upper cache asynchronously, the updates are sent via unicast to every single cache at the lower level without significantly reducing the load on the caches. If an upper cache multicasts document updates to all synchronized lower caches the model changes from a *receiver-initiated* caching scheme to a *push-caching* scheme. Pushing document updates in advance reduces the Update misses.

The caching-multicast cooperation could work like this:

1. A final Web server is the only one that certainly knows when a document has changed. Only popular and changing documents should be pushed to avoid a lot of overhead on the multicast routers.
2. Every time that a popular document changes, the Web server can take the decision to multicast the document update towards all the national caches.
3. The national caches themselves forward the document update to all regional caches.
4. The regional caches keep track of which documents are popular for their children. Based on these statistics the regional caches decide to keep the document or to remove it, performing a geographical filtering.
5. The regional caches that have interested receivers in that document update, will forward it towards all institutional caches.
6. The institutional caches will do the same process as the regional caches.

The drawback of this approach is that some caches may receive a document update for which they are not interested. One possible solution would be to announce the document update previously on a signaling multicast group. All interested caches would join the multicast group and receive the corresponding document update leaving the group later.

6 Conclusions

Caches need additional resources and placement in the network. The placement of caches will eventually happen because ISPs are very interested in saving network bandwidth and in reducing the latency to their receivers. Caching is an application hop-by-application hop. When a document is stored at lower level caches, further requests for the same document can benefit from higher network capacities. On the other hand, multicast is still in its infancy. Multicast is an end-to-end solution and therefore is limited by the most restricted bottleneck on the whole path from the source to the receivers.

In this paper we find that except for popular and fast-changing documents, a caching hierarchy has the lowest latency delivery time. For fast-changing popular documents the caches at the higher levels become overloaded and the completion time experienced by the receivers is higher than using a multicast distribution. Additionally, the time to obtain the first packet of a document is higher on a caching distribution than on a multicast distribution.

Concerning the bandwidth, we have also some results not presented in this paper showing that for very popular and fast-changing documents the bandwidth used on a caching distribution is higher than on a multicast distribution, especially when the document size increases.

The use of caching and multicast together gives better performance results (latency, bandwidth) than each of them alone. We claim that due to the varying nature of Web documents both caching and multicast should be implemented on the distribution of Web documents on the Internet.

7 Acknowledgments

The support of P.Rodriguez by the European Commission in form of a TMR (Training and Mobility for Researchers in Europe) fellowship is gratefully acknowledged. 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] A.Rousskov, "On Performance of Caching Proxies", , NoDak, February 1998.

[2] 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.

[3] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT)", In *Proceedings of SIGCOMM'93*, pp. 85–95, San Francisco, CA, USA, October 1993, ACM.

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

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

[6] R. Clark and M. Ammar, "Providing Scalable Web Service Using Multicast Delivery", In *Proceedings of the IEEE Workshop on Services in Distributed and Networked Environments*, Whistler, Canada, June 1995.

[7] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei, "The PIM Architecture for Wide–Area Multicast Routing", *IEEE/ACM Transactions on Networking*, 4(2):153–162, April 1996.

[8] M. Doar and I. Leslie, "How Bad is Naïve Multicast Routing", In *Proceedings of IEEE INFOCOM'93*, volume 1, pp. 82–89, 1993.

[9] 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.

[10] H. Eriksson, "MBONE: The Multicast Backbone", *Communications of the ACM*, 37(8):54–60, August 1994.

[11] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, et al., "RFC 2068: Hypertext Transfer Protocol — HTTP/1.1", January 1997.

[12] A. Gove, "Stream on: RealNetwork isn't about to make multicast video a mass medium", *The Red Herring*, November 1997.

[13] J. Gwertzman and M. Seltzer, "The Case for Geographical Push Caching", In *Proceedings of the Fifth Annual Workshop on Hot Operating Systems*, pp. 51–55, Orcas Island, AW, May 1995.

[14] V. Kumar, "The MBONE FAQ", Collection of Information about MBONE, January 1997.

[15] R. Malpani, J. Lorch, and D. Berger, "Making World Wide Web Caching Servers Cooperate", In *Fourth International WWW Conference*, Boston, Dec 1995.

[16] J. Nonnenmacher and E. Biersack, "Asynchronous Multicast Push: AMP", In *Proceedings of ICCC'97*, pp. 419–430, Cannes, France, November 1997.

[17] V. N. Padmanabhan and J. Mogul, “Improving HTTP Latency”, In *Second World Wide Web Conference ’94: Mosaic and the Web*, pp. 995–1005, October 1994.

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

[19] P.Rodriguez, E. W.Biersack, and K.W.Ross, “Latency analysis for a Caching and Multicast distribution”, , Institut EURECOM, 2229 route des Crêtes, B.P. 193, 06904 Sophia Antipolis Cedex, FRANCE, March 1998.

[20] P. Rodriguez and E. Biersack, “Continuous Multicast Distribution of Web Documents over the Internet”, *To appear in IEEE Network Magazine*, March 1998.

[21] K. W. Ross, “Hash-Routing for Collections of Shared Web Caches”, *IEEE Network Magazine*, 11, 7:37–44, Nov-Dec 1997.

[22] N. G. Smith, “The UK national Web cache - The state of the art”, *Computer Networks and ISDN Systems*, 28:1407–1414, 1996.

[23] R. Tewari, M. Dahlin, H. M. Vin, and J. S. Kay, “Beyond Hierarchies: Design Considerations for Distributed Caching on the Internet”, , The Univertisy of Texas at Austin, February 1998.

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

[25] D. Wessels, “Squid Internet Object Cache: http://www.nlanr.net/Squid/”, 1996.

[26] 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.