International Journal of Computer Science & Applications
Vol. II, No. II, pp. 5 - 21

The impact of Proxy caches on Browser Latency

Andrzej Sieminski
Institute for Applied Informatics
Technical University of Wroclaw, Poland
andrzej.sieminski@pwr.wroc.pl

Abstract:

Proxy caches had become popular in the mid 90's. At that time the two main reasons for using proxies were: security of Internet access, the limitation of global network traffic and the speeding up network connections. The first two issues as important as they were before. The question is, however, whether proxies could still speed up the transfer rate for an individual user considering the fact, that in the last couple of years the average Internet transfer rate of an individual user has risen several times. The paper considers different types of proxy actions and studies in what way they influence the browser display time. It discusses also acceptable loading times and the scope of cacheable objects. The paper introduces the CF Cacheability Factor which estimates the susceptibility to caching of individual WWW objects and whole Web sites as well. The CF could be used also to describe the browser cache – proxy cache cooperation.

1. Introduction

Memex, the very first concept similar to the World Wide Web (WWW) was described as long as 60 years ago [3]. However it was not until the 1990's that the concept was finally designed and implemented. At that time the structure the WWW was remarkably simple: it could be regarded as a collection of web servers and browsers. The concept of multimedia documents that could be linked to each other over a network spanning the whole globe was so attractive, that since mid 1990's we have witnessed an unprecedented growth of the WWW, both in the terms of content and usage. As a result the simple structure had to evolve. Now there are many additional levels such as LAN and Internet proxy servers, Content Delivery Networks (CDN's), surrogates, mirror sites, WEB accelerators and Notification Systems [7]. Probably the most widely used are proxy severs.

A proxy is a trusted agent that can access the Internet on behalf of its users. There are two types of proxies: LAN based and Internet based. The proxies of the first type operate on the edge between LAN and Internet. Only users of the LAN network could get access to the proxy server. The second type proxies operate in Internet at large. They could be used by users from a specific region or even, without any restriction, by all users. The two types differ not only in localization and accessibility but also in the functions that they offer. In the case of LAN proxies the functions include:

- providing access to Internet from within a local network;
- controlling the access to network recourses
- protecting individual users from network attack.
- reducing necessary bandwidth.

The main aim of Internet proxies is to reduce the long distance network traffic. Caching of popular objects occurs in both types of proxies. Proxies are located closer to a user then WWW servers, so it was generally believed, that they should significantly reduce the download time of WWW objects [20]. The paper analysis under what conditions such an assumption is still justified. In the past the proxies were subject of

intensive study but usually the attention was focused on studding the byte or object hit rate. The metrics refer to the proportion of bytes or objects served from the cache buffer and they could be used to estimate the reduction of latency. The studies of download time reduction are much more difficult and are less common [11, 8].

The paper starts with a discussion of acceptable and actual download time of Internet pages. The conclusion is, that the increasing of network bandwidth alone could not decrease significantly the time. This suggests the usage of intermediaries such as proxies. The next section deals with the scope of cacheable data. It starts with a brief description of formal cacheability as defined by HTTP/1.1 protocol and continues with an assessment of the cacheability of Internet objects at large and a detailed analysis of some popular web sites. The analysis describe also how webmasters use and sometimes misuse the HTTP guidelines. The section 4 discuses other papers on latency reduction due to the proxy deployment. The next chapter deals with general conditions that must be fulfilled if a proxy is to speed up the transmission rate. The process of proxy caching itself is described next. Part of the chapter is a study of proxy log files. The study answers following questions: how often the different operations are actually performed and how do they contribute to the overall download time. In order to analyze he impact of proxies we have to analyze the cooperation between a browser cache and a proxy cache with respect to the properties of web sites. The problem is addressed in the 7th section. The paper ends with a discussion of the achieved results and presents areas of future work.

2. The Latency Problem

Latency is defined as the delay between a request for a Web page and receiving that page in its entirety. The latency problem occurs when users judge the download as too long. Unacceptable latency does not only adversely effects user satisfaction. Web pages that are loaded faster are judged to be significantly more interesting then their slower counterparts [4]. Although the problem is important for all pages, in the area of e-commerce it is absolutely crucial. Too long download time prompts users go to different sites with similar offer. The revenue lost that way was estimated at over $300 ml in 2000 in the USA alone [23].

The work on the Advanced Research Projects Agency Network (ARPANET) – the global Internet's progenitor had started in 1960's. The initial motivation for both ARPANET and Internet was resource sharing, file transfer, remote login applications and email. The WEB was not developed until several years later. Therefore the Internet TCP/IP protocol is not well suited for the kind of responsiveness that is expected by users of the WEB. In Internet the communications are on the best effort basis. If a packet didn't make it to the final destination, it would shortly be retransmitted from the source. Such a way of operation is not conducive to transfer rate, it fevers much more reliability in unpredictable situations. It seems that HTTP spends more time waiting than it does transferring data [18].

2.1. Acceptable Levels of Latency

Studies on human cognition revealed that the response time shorter then 0.1 second is unnoticeable and the delay of 1 second matches the pace of interactive dialog. On the average a man executes elementary operations, so called "unit tasks" at the pace of 6 tasks per minute [4]. It could suggest that a latency of over 10 seconds disrupts the unit task and results in user disorientation and reduced performance. The result corresponds with the 8 second latency limit declared in the widely cited report [23]. Experiments described in [1] indicate that there are more factors that influence acceptable levels of latency. Proficient computer users are less patient then computer novices. The tolerance for delay is decreased as the length of time a user spends interacting with a site increases. The most surprising fact is the effect of incremental loading – it could six fold increase the user tolerance, see Table 1.

Table 1 Rating of latency [1]

Rating	Regular Latence	Incremental Latency
High	0-5 sec	0-38 sec
Average	>5	>39 sec
Low	>11 sec	>56

The standard browser feedback in form of progress bars does not provide users enough information to keep them busy. During incremental download they first receive the banner of the next page right after the click, followed by text and the end graphics.

2.2. Actual data

In Internet we witness two opposing trends. On the one hand there is the ever increasing transfer rate of both user connections and the increasing of the bandwidth of the skeleton network. On the other hand the internet pages become more and more complicated and therefore more "fat", their size steadily increases. The growth of transfer rate is much more spectacular e.g. the most popular in the early 90's 28kB modem was replaced by a large variety of connection types, see Table 2.

Tab 2 The transfer rates for different connection types

Connection Type	Slow	Normal	Maximum
Modem 33k6	<2,800 Bps	~3,300 Bps	3,733 Bps
Modem 56k	<4,300 Bps	~5,300 Bps	6,222 Bps
ISDN 64k	<5600 Bps	~6,400 Bps	7,111 Bps
Cable	<10,000 Bps	~17,500 Bps	by provider
ADSL	<12,500 Bps	~25,000 Bps	~750,000 Bps
Ethernet 10Base-T (10 Megabit/sec)	<75,000	Bps ~200,000	Bps ~1,000,000 Bps

Therefore one would expect that the 8 second rule should easily satisfied. This is unfortunately not the case [5]. The user perceived latency is subject to many factors: WWW server work load, Internet flash crowds [10] and the slowest link all the way from a browser to a WWW server.

The statistics published on Internet clearly indicate that the average speed increases but at a pace far lower then the upper bound of technical specifications. The Table 3 shows the changes in the Internet transfer rate on international connections in 8 selected countries and world wide during the last 4 years. The data are obtained from the [W5] site, probably the one of the most popular websites devoted to speed measurement. According to the data, in most countries (except France) the increase is not substantial and world wide we witness even a decrease of transfer rate. The internal transfer rate shows a increases more steadily but it does not match growing number of users, sites and complexity of pages.

Table 2. External Transfer rate in KB/sec

Country	Year			
	2005	2004	2003	2002
Brasil	14,18	14,29	12,72	11,32
France	30,67	22,29	22,28	19,79
Germany	21,30	16,50	16,73	18,96
Japan	27,07	28,49	23,75	20,04
Mexico	21,30	22,16	22,62	20,04
Poland	16,33	16,29	13,22	10,24
UK	26,68	24,33	22,26	25,03
USA	34,33	37,23	30,63	31,48

World	18,72	18,61	18,89	24,78

The technological improvements alone clearly could not solve the latency problem. Internet is based on TCP protocol and the protocol is not well suited for the transmission of small objects that make the bulk of Internet traffic. The maximum theoretical transfer rate on cross Atlantic connection for 1 kilo byte transfers is below 4KBytes per second. It means that the increase in backbone performance will not have an substantial impact on small data transfers [14] and the trend will continue in the foreseeable future.

Most of the data transmitted on Internet uses the HTTP protocol. With the increasing role of multimedia (internet TV, radio or TV on Demand) the streaming data will increase their share in the volume of transmission. The impact on the HTTP transfer rate will be even more noticeable as their protocols put much emphasis on transfer rate. The increased throughput of the Internet will be consumed mostly by the multimedia and other protocols.

The users are not generally satisfied with the current level of the QoS. It is the reason for the commercial success of the CDN (Content Delivery Networks) [19] companies like Akamai [W2]. The companies with substantial financial resources could buy the acceptable latency. This is not a solution for servers but not for users.

3. Cacheability of WWW objects

The Internet transfer rate is effected by a number of reasons: network workload and throughput, flash crowds, WWW server workload. They are beyond the control of an individual user. The caching of WWW objects offers the possibility to reduce the unacceptable latency levels. This section describes what and for how long could be cached

3.1. HTTP guidelines

The HTTP/1.1 protocol specifies in a very detailed manner what data could not be cached [W3]. Roughly speaking the uncacheable data falls in one of the following categories [22]:

- A browser has issued a request for a dynamic page. Dynamic pages are generated by the server each time a request is made and are therefore not cacheable. They could be identified by the presence of special characters within the URL: ”?”, ”=”,”/cgi-bin/”, ”.cgi”, ”.pl”, or ”.asp”.
- A browser has used one of uncacheable methods. In HTTP 1.1, there are seven different methods but only two of them can be cached: GET and HEAD.
- A browser had received an uncacheable HTTP Status Codes. The HTTP 1.1 defines cacheable, negative cacheable and uncacheable status codes. Negatively cacheable codes are less restrictive then uncacheable codes. They are used when for a short amount of time, a cache can send the local copy without fetching it from original Web server. The most popular negative status code is Service Unavailable. It is save to assume that for few seconds the service will not start to operate.
- A browser had received an uncacheable response headers. These headers are used for cache control, indicate a necessary user authorization or are used with cookies.

3.2. Scope of cacheable objects

There were many approaches to estimate the scope of cacheable data. Interest in the subject is shown mainly by the caching community. The estimate is also the part of models of the changeability of Internet. Such models are necessary o estimate the

required the performance of page indexing applications used by search engines such as Google.

The obtained results differ to some extend depending on the scope of data and the methodology used. The most simple case the log of a proxy server is analyzed. The URLs that were more often served from the cache then from the WWW server are treated as cacheable. The method gives us only a rough estimate as it does not take directly into account the cache replacing algorithm and effect of the cooperation of caches [21]. It could be described as a practical cacheability. It differs depending on a type of object and is equal to 91% for pictures and 37% for HTML text, 76% being the average.

A more precise study treats the log data only as a source of URLs and then the headers of the objects are downloaded (proxy logs do not contain the headers). The headers are interpreted according to the HTTP specification to determine the cacheability of an accompanying object [22]. Not surprisingly the results indicate a greater portion of cacheable data – about 81%.

The most accurate way requires the analysis of the actual changes in the content of the objects [2]. Objects are loaded periodically. An object is classified as a dynamic if each time a different content was fetched. The results indicate that only as little as 4% of objects are in fact dynamic. The result has however a limited usefulness to caching as the time span between consecutive test to long - on the average 12 observations in 37 days. This appropriate for page indexing but not for caching. It should also be stressed that the scope of under the study was limited to the pages retrieved by Informat system.

3.3. Caching in HTTP

The overall majority of pictures and a significant part of HTML object could be cached. The HTTP/1.1 [9] includes two mechanisms to promote the caching of data:

- expiration mechanism that enables the cache to serve an object directly from the cache without any communication with the WWW server and
- validation mechanism that reduces network bandwidth requirements as on the average shorter objects transmitted from the WWW server.

The first mechanism is used for fresh objects - objects that have not yet expired. Servers could specify explicit expiration times using either the Expires header, or the max-age directive of the Cache-Control header. Unfortunately the WWW servers that provide explicit expiration times are relatively rare – almost 90% of all responses do not contain either Expires nor max-age. This limits the scope of cacheable data and therefore the HTTP/1.1 specification allows the caches to use heuristic expiration times, employing algorithms that use other header values. The most common algorithm uses the Last-Modified time to estimate a plausible expiration time. It is usually set to a fraction of the age of an objects. Typical settings of this fraction are from 10% to 20%.

The validation mechanism is used for stale objects. Such objects could also, in some cases, be delivered form the buffer. The cache first has to check with the origin server to see if its cached entry is still usable. This is accomplished with conditional HTTP methods. The methods use object validators: Last-Modified object-header or the ETag response-header. If the cache and the WWW server validators match, then the WWW server responds with a special status code (usually, 304 (Not Modified)) that is no usually many times shorter then the object body. Otherwise, it returns a full response. The Last-Modified object-header is especially important for caching as it could be used by both mechanisms.

4. Related work

Proxy caches promise the reduction of network traffic, user latency and additional non-technical features such as content filtering. There are a number of metrics to evaluate the buffering performance: object hit rate, byte hit rate, latency reduction, saved bandwidth, CPU performance, disk performance [14]. The vast majority of papers

dealing with the proxy evaluation concentrate upon the object and byte hit rate and analyze how different replacement algorithms influence these basic metrics. It should be stressed that these metrics do not translate directly into the latency reduction.

The upper bound of latency reduction due to proxy deployment was estimated by Kroeger [11]. Unlike the analysis presented in Section 5 his studies presume that a proxy server is always present, they do not include proxy overhead and assume a fixed browser-proxy latency. Under those assumptions the estimated best case latency reduction was 40%. The result was verified by a simulation. In that case the observed reduction was much lower ranging from 22% to 26%.

A more comprehensive study was described in [8]. It included some factors omitted in [11] but still the use of a proxy server was obligatory and it covered two environments: one with browser-proxy connection via a slow modem and second with an Ethernet connection. As a result the slow modem connection was less influenced by a proxy (latency reduction 3%) then an Ethemet connection (latency reduction by 7%). The paper advocates strongly TCP connection caching instead of object caching in favor of object caching.

A proxy splits the TCP connection. The resulting reduction of RTT (Round Time Trip) increases the effective bandwidth due to operation of TCP protocol. These are low level optimizations and are included in the formulas presented in section 5.

Some researchers attempted to evaluate the latency reduction in specific environments. M. Person et al. studied e.g. different proxy configurations for streamlining Internet traffic in New Zealand [13] . The country is geographically isolated and much of Web traffic has to cross the Pacific Ocean which results in RTT exceeding 120 milliseconds. Another such example is the use of proxies for satellite links. Such links are characterized by even higher RTT's (often in the order of seconds) and extremely high bandwidth capacity. This prompts to place proxies on both ends of distinctive links and modify TCP behavior [6]. While offering potentially substantial latency reductions those approaches are not relevant to the paper as it deals with the mainstream of Web traffic.

5. Conditions for latency reduction

The paper presents a simple model that could be used to evaluate all possible browser-proxy configurations.

In what follows we shell compare two types of internet connection:

- TypeA (without a proxy server) and
- TypeB - with a proxy servers, see Figure 1.

Let *DtA*(X) and *DtB*(X) denote the download time of an object X using the TypeA or TypeB connections respectively. A proxy server decreases the latency of loading an object X if the following inequity holds.

$$DtA(X) = \frac{Xs}{Tbw} > DtB(X) = \left(\frac{Xs}{Tbp} + \frac{Xs}{Tpw} \right)$$

where:

Xs – size of the object X

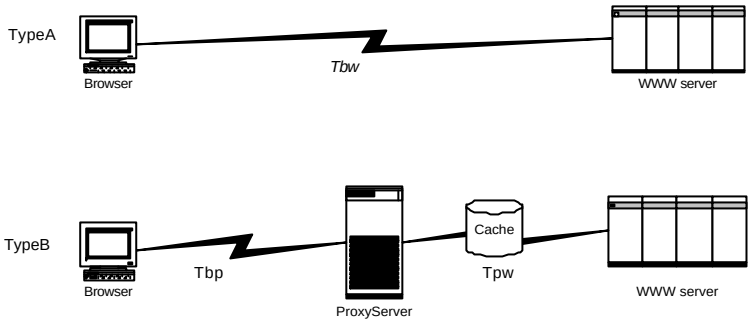
Tbp – Transfer rate browser - proxy

Tpw – Transfer rate proxy served objects

Tbw – Transfer rate browser - WWW server

The *Tpw* is the average transfer rate of all objects delivered by the proxy server. Some of them are served directly from the proxy cache, while others are must be loaded from a WWW server.

Figure 1 TypeA and TypeB Browser access to Internet



TypeB connection offers faster transfer then TypeA if *Tbw* is satisfies the condition:

$$Tbw < \frac{Tbp * Tpw}{Tpb + Tpw}$$

The browser-proxy connection takes time, so even if a browser can has a slower internet connection then a proxy it is still possible that the TypeA could be faster then TypeB. The effect diminishes with increasing values of Tbp, see Table 3.

Table 3. Minimal values for *Tbw* for which TypeA is slower then TypeB connection.

Tbp	*Tpw*	min *Tbw*
10	10	5,00
100	10	9,09
1 000	10	9,90
1 000	20	19,61
1 000	50	47,62
1 000	100	90,91

For any object X the gain factor *G(X)* defined as follows:

$$G(X) = \frac{(DtA(X) - DtB(X))}{DtA(X)} = 1 - \left(\frac{Tbw * (Tbp + Tpw)}{Tbp * Tpw} \right)$$

represents the impact of introducing a proxy server. Let us assume that a browser is connected to a proxy using a slow modem. In such a case *Tbp=Tbw* and as a result the introduction of a proxy in fact increases the latency. In that case the value of G(X) is negative and is equal to:

$$G(X) = 1 - \frac{Tbw * (Tbp + Tpw)}{Tbp * Tpw} = - \frac{Tbp}{Tpw}$$

The introduction of a proxy server is justified if the proxy has a significantly faster transfer rate then a browser, that is: *Tpw*=k**Tbw*, where k>>1. The introduction of a proxy server shortens transmission time if the transfer rate browser is proxy is fast enough, more precisely if:

$$Tbp > \frac{k * Tbw}{(k - 1)}$$

In such a case the gain factor is equal to:

$$G(X) = 1 - \frac{1}{k} - \frac{Tbw}{Tbp}$$

For larger values of k the transfer rate browser-proxy must be almost equal to the transfer rate proxy-WWW. The condition could be easily satisfied for all LAN based proxies but certainly not for Internet proxies.

This above conclusions correspond well with the findings presented in [8] where a latency reduction of 3% and 7% for a slow 28.8 kbps modem and 10Mbps Ethernet connection respectively were reported. The above results refer to objects that are fetched from a proxy server. In practice many objects are supplied by the browser cache. A discussion of this problem is presented in Section 6.4.

6. Proxy caching

A proxy cache is obliged to respond to a request with the most up-to-date response held by the cache that is appropriate to the request. A singe proxy server has many users. Its buffer is practically always full. Therefore one needs a replacement algorithm that decides which objects to keep and which to delete from cache. The problem is well defined, has clear efficiency measures (e.g. object or byte hit ratios) and is important from the practical point of view. It is no wonder, that in the early days of caching much work has been done on that area and many replacement algorithms were proposed e.g.: FIFO, LFU (Least Frequently Used) , LRU (Least Recently Used), LRU-Threshold, Lowest_Latency_First and GreedyDual Size [14]. The algorithms based on the LRU principle are considered to be the best and it is rather unlikely, that a new algorithm would be published that has significantly better parameters.

Uncacheable object are fetched form the original WWW servers. Cacheable objects are delivered from the cache with or without validation or from the WWW servers if they are not to found in the cache.

6.1. The actions

The basic idea of caching is simple on its own but the actual actions taken by a proxy cache are rather complicated. The is due to the fact that a proxy has to implement the HTTP protocol. The HTTP versions 1.0 and 1.1 specify in a very detailed manner what objects could be cached, for how long they could be regarded as fresh and how to validate them when they have expired. The exact type of action depends therefore on five factors:

- Cacheability of the requested object,
- Whether the object is already cached or not.
- Whether the cached object was stale or fresh in browser cache.
- Whether the cached object was stale or fresh in proxy cache.
- Type of user request: regular or conditional.

The Table 4 summarizes the actions of Squid – probably the most popular proxy cache [W6]. The actions relate directly to the response code delivered by the proxy server. Although they were defined for a particular proxy type they have a general validity as they cover all types of HTTP request and responses and all possible locations and freshness statuses of WWW objects.

Tab. 4 Actions of the Squid proxy browser.

Action Code	Actions' description
RegHIT	Regular request, the object in the proxy cache was valid and was delivered from the it.

StaleHITb	Regular request, the object in the proxy cache was stale but was accepted after verification at the WWW server. The cached object was delivered form the proxy.
StaleHITh	Conditional request, the object in a browser cache was stale but it was accepted after verification at the WWW server. Only short confirmation message was sent to the browser.
ImsHITb	Conditional request, the object in the browser cache was stale. A fresh object was delivered from the proxy cache.
ImsHITh	Conditional request. The browser regarded the object as stale. It was however still fresh, according to the data kept in the proxy server. Only short confirmation messages was sent to the browser.
MemHIT	Regular request, a valid copy of the object was in the cache mamory and was sent from it, thus avoiding disc access.
RegMISSb	Regular request, the object was not in cache, the object is delivered from the WWW server.
ImsMISSh	Conditional request. The browser regarded the object as stale. It was not kept in the proxy cache. The verification at the WWW server confirmed the freshness of the object. Only short confirmation messages was sent to the browser.
ImsMISSb	Conditional client request, The object was in the proxy buffer was stale. A fresh object from the WWW server was sent to the browser.
NoCacheb	No_cache request, the object had to be sent from the WWW server.
NoCacheh	No_cache client request, The browser object was accepted after verification at the WWW server and a confirmation message was send to the browser.

The actions implement fully the cache supporting headers defined in the HTTP protocol. The only extension is the MemHIT action which fetches the most popular objects from main memory not from disk. This is only a performance measure.

The actions names containing "HIT" involve sending data from the proxy cache, but in the case of StaleHIT actions the objects kept in the proxy cache must be verified at the WWW server. All actions with names ending with "b" send the whole body of the object that is its contents. The actions with h and the end of their names send only a header that confirms the validity of on object in the browser cache. The confirmation header is relatively short, it contains ca 400 bytes.

6.2. Log analysis

The Table 5 shows the frequency of use, byte transfer rate in Bytes per millisecond and average object size for all actions presented in the previous section. The raw data were obtained from NLANR (National Laboratory for Applied Network Research). The NLANR runs a network of Internet proxy caches using the Squid cache engine. If a given proxy server in the network does not have a requested object then it will first consult other proxy servers connected to it as parents or siblings before sending the request to a WWW server. The logs from the last 7 days are published [W1] and are free for research purposes.

The selected logs contain data collected from two proxy servers bo1 (Boulder, Colorado) and pa (Palo Alto, California) on the 14th. of May, 2002. Each of the logs describes more then 200 000 request.

Tab. 5 Analysis of LNANR log data from two proxy servers

Action Code	Frequency		Avg. Size		Avg. Tran. Rate	
	pa	bo1	pa	bo1	pa	bo1
RegHIT	22,78	4,16	20027,26	65665,49	8,23	7,19

StaleHITb	6,54	5,96	6503,54	8047,80	5,90	2,81
StaleHITh	3,71	7,84	265,72	268,71	0,41	0,56
ImsHITb	0,25	0,25	56946,81	44579,17	17,73	8,24
ImsHITh	15,13	10,27	269,06	281,29	2,68	4,87
MemHIT	0,42	0,20	2902,37	2502,67	21,00	138,19
RegMISSb	19,71	57,70	11830,80	15184,09	2,48	4,12
ImsMISSh	1,13	7,45	283,04	278,04	0,48	0,32
ImsMISSb	1,36	1,95	10828,01	11232,82	2,40	3,29
NoCacheb	1,19	6,00	4365,18	3385,57	2,59	6,11
NoCacheh	0,65	4,87	282,16	279,60	0,98	0,94

The caches operate at a national level and their clients are lower level caches not individual users. Such a way of operation decreases the amount of objects served from cache, many popular objects are loaded from browser or lower level buffers. The object hit ration Proxy caches operating closer to a browser could well surpass the hit ratio form. The pa server operates as a single node, closer a an individual user and has an object hit ratio equal to 48,83% whereas bo1 achieves only 28,68%. The average sizes of objects are similar with one notable exception: the size of objects served directly from the cache using the RegHIT action. The average size of such objects is more then three times greater for bo1 cache then the pa cache. This means that lower level caches have buffered smaller objects by their own. Larger objects were considered not worth caching.

It looks like the bo1 cache had a faster internet connection. Many actions that require contact with a WWW server (StaleHITh, ImsHITh and all MISS actions) are faster then in the case of pa cache server.

The most striking feature of the data is that the speed advantage offered by caching is small if any, see table 2. In the case of bo1 server the validation of the object body kept in a proxy buffer took so long that it was faster to fetch the data directly form the WWW server and as a result the transfer rate for NoCacheb surpasses the transfer rate for StaleHITb. It is not true for the pa proxy that has a slower Internet connection. Proxy server could significantly increase the transfer rate only when the object is kept in main memory. In the case of both of the proxies the scope of memory fetched objects is marginal. It would suggest that a hope expressed in [14], that a properly configured proxy server could serve a substantial amount of object from main memory does not occur in practice.

7. Caching in Practice

The section deals with the influence of the mentioned above factors on an individual user. A proxy server is only one element of the complex structure of Internet middleware. First we study the modes of a browser cache and its interaction with a proxy cache. Next we analyze the cacheability of typical Internet sites.

7.1. Browser cache

Due to the declining cost of disk memory the browser caches have now sizes that could easily store all objects downloaded by an individual user during the period of several days. The importance of replacement algorithms so crucial for web caches is thus negligible. A browser cache delivers objects to a single user and therefore is not required to fulfill the requirements of the HTTP protocol. This is reflected in the modes of operation of a browser cache. They differ in the situations when a decision to check the validity of an object is taken. The validity could be checked:

- on each visit on a page - all objects are always verified using usually the If_Modified method;
- only once - the check is done only on the first visit on a page during a single browser session. All subsequent visits to such a page use cached objects.

- automatically – the cache checks header information like a proxy but also an attempt is made to take into account the actual changes of an object.
- never – the check is never done, thus allowing the user to browse offline the cached pages.

The third mode is by far the most popular. It implies work similar to that of an Internet proxy cache. In what fallows we assume that the browser uses the third method.

7.2. Analysis of selected Web sites

The sections presents a detailed analysis of the cacheability of selected Web sites. This gives us the possibility to see how the Webmasters use the headers offered by the HTTP/1.1 protocol. The sites were divided into 5 groups:

- Internet shops and auctions (Shops):
  - allegro (http://www.allegro.pl),
  - Aby (http://www.aaaby.pl/0,0.html),
  - Ebay(http://www.ebay.co.uk/).
- Official City Sites (Cities):
  - Wroc (http://www.wroclaw.pl/m/),
  - Wwa (http://www.warszawa.p l/),
  - Londyn (http://www.london.gov.uk/).
- Software Companies (Soft):
  - Prokom (http://www.prokom.pl/pl/),
  - Softbank (http://www.softbank.pl/),
  - Borland (http://borland.com/).
- Internet Newspapers (News):
  - Rzepa (http://www.rzeczpospolita.pl/),
  - Wp (http://www.wp.pl/),
  - CNN (http://www.cnn.com).
- Internet Journals (Journals):
  - ChipPl (http://www.chip.pl/),
  - ChipDe http://www.chip.de/,
  - Byte (http://byte.com/).

The selection of sites was arbitral but all of them are popular in Poland, have evolved over a period of several years and have the backing of large companies. In order to mimic the browsing behavior of an user a group of 4 pages from each site were collected. Each group had a structure shown on the Figure 2.

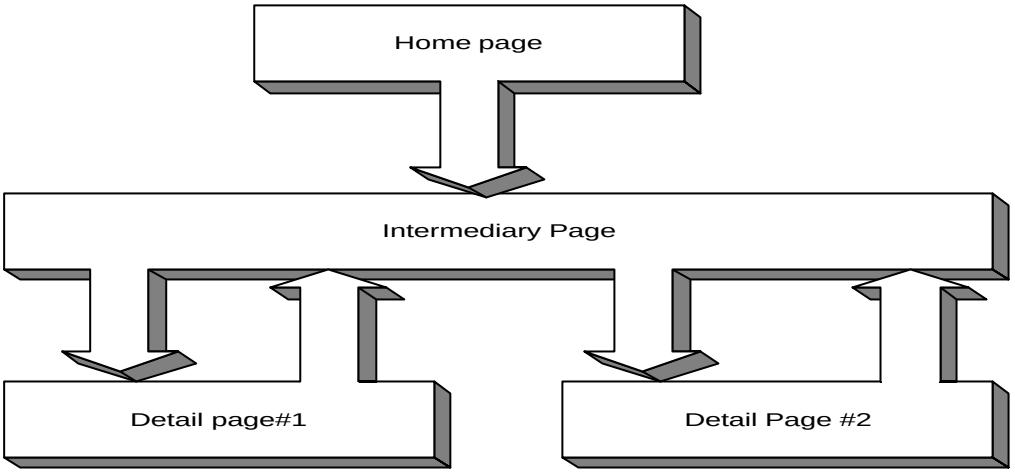
It is assumed that the pages are visited in the following order:

Home Page⇒Intermediary Page ⇒Detail Page#1⇒Intermediary Page ⇒Detail Page#2

The Home Page and Intermediary Page are navigational pages while Detail Page #1 and #2 are content pages [12]. The above sequence of pages is called a test sequence. The objects (files) making up the sequences were divided into:

- text – file extensions: HTML or HTM
- graphics – file extensions: gif, jpg, png
- program – file extensions: css, jsp, php, js

Figure 2 The structure of w test Website.



The availability of the headers is shown in the Table 6. The pages were analyzed by an application available on the WEB page [W4]. The codes used in the table have the following meaning:

- Cookie – all types of cookie herders
- Age – the headers MaxAge or Expires with a date later then the dawnload time
- LM – the LastModified Header
- NoCache – all types of cache or store headers

The table 6 shows some remarkable differences in the cache friendliness of different sites groups. The most cache friendly is the group consisting of shops. They provide the most detailed information about the changeability of objects. The cookies are attached only to HTML objects. On contrast in the News group cookies come with half of all picture objects, thus preventing their caching. Such a way of constructing of a page could be described as a misuse of HTTP/1.1 guidelines.

To study the objects lifespan a cacheability factor CF was presented in [16]. The factor for object x is calculated using the headers supplied with object x and a given refresh rate R:

$$CF(x,R) = \frac{(PLn(x,R) \quad - \quad R)}{PLn(x,R)}$$

where:
x – the WWW object under study
R – refresh rate, the number of seconds between consecutive reading of the object
PLn(x,R) – Normalized Predicted Lifespan

$$PLn(x,R) = \begin{cases} PL(x) & for\ \ PL(x) > R \\ R & .in\ other\ \ case \end{cases}$$

where PL(x) is the predicted Lifespan of x calculated using the HTPP headers.

Table 6 Availability of cache controlling headers.

Site		Header			
Type	Object type	Age	LM	No_Cache	Cookie
Total	HTML	21%	7%	4%	44%
	Pic	28%	96%	1%	12%
	PRG	34%	14%	6%	14%
City	HTML	0%	0%	0%	20%
	Pic	1%	85%	0%	0%
	Prg	0%	90%	0%	10%
News	HTML	33%	17%	8%	67%
	Pic	4%	97%	10%	50%
	Prg	29%	62%	4%	54%

Shop	HTML	33%	17%	0%	83%
	Pic	0%	99%	0%	0%
	Prg	4%	96%	1%	4%
Mag	HTML	33%	0%	3%	42%
	Pic	33%	96%	0%	4%
	Prg	51%	54%	22%	12%
Soft	HTML	3%	27%	0%	0%
	Pic	1%	98%	2%	0%
	Prg	0%	96%	0%	0%

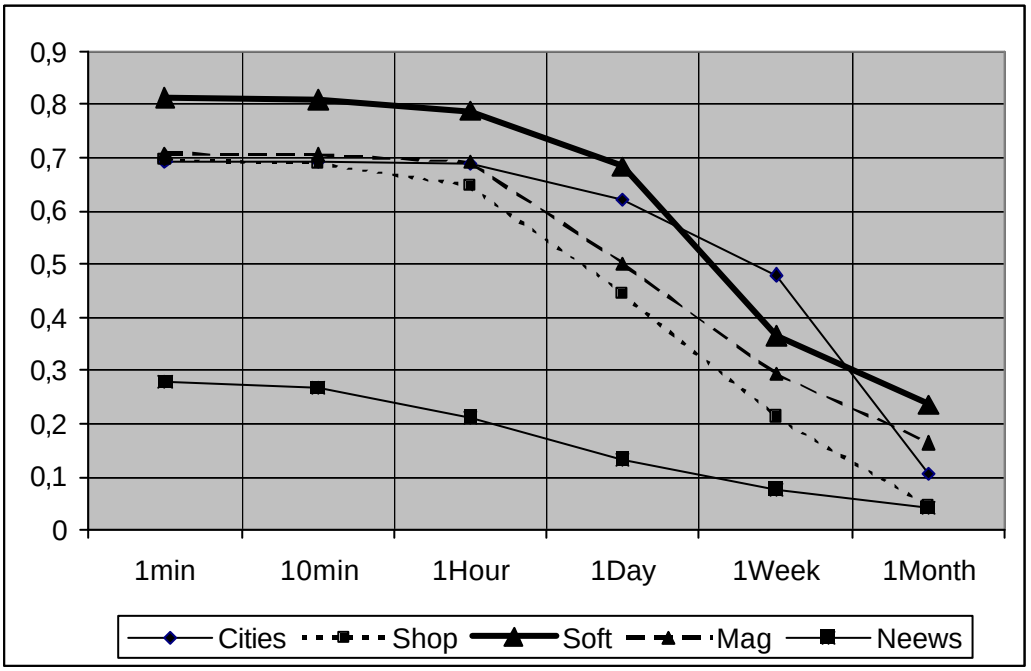
In the rest of the paper following values of R are listed in Table 7. The CF(x,R) could be interpreted as a probability that an object would be loaded from a buffer and not from a WWW server. The Cacheability Factor is defined for a whole page or a sequence of pages as a sum of CF for all individual objects weighted by the object size.

Table 7 Selected refresh rates.

Code	Interpretation
RMin	Skiping a page
R10m	Reading a page
RHour	Reurn to a page during the same session.
RDay	Return to a page on daily basis.
RWeek	Regular bur not frequent returning to a page.
RMonth	Sporadic return to a page.

The Figure 3 shows the values of the Cacheability Factor for the selected refresh rates for test websites. The Tables 8 to 10 present the total values for each website and for each object type.

Figure 3 Cacheability Factor for websites



The data presented in these tables clearly indicate that there are huge differences in the susceptibility to caching of different web sites. The news websites change most often and therefore their CF is the lowest. The highest values of CF achieve websites of Software companies. This the combined effect of cache awareness of their web masters and the relative stability of presented contents. The worst results achieve the cites web sites. Their content is undboutfully by far the most stable in the test data but this is not reflected by the relatively low values of CF. The analysis of CF values for the shops web sites reveal that their web masters have done much to promote caching. Although they consist of dynamically created pages they CF values for RMin, R10Min and RHour remain almost stable. This facilitates browsing sessions.

Table 8 The Cacheability Factor for PRG type objects

Refresh R.	News	Shop	SoftCom	Cities	Mag
RMin	0,8064	0,954	0,9246	0,8983	0,7049
R10Min	0,7952	0,9448	0,9174	0,898	0,7029
RHour	0,7615	0,8936	0,8776	0,8966	0,6915
RDay	0,4185	0,3583	0,6952	0,8584	0,4783
RWeek	0,3198	0,0728	0,3423	0,6391	0,178
RMonth	0,0000	0,0000	0,0000	0,0000	0,0152

Table 9 The Cacheability Factor for PIC type objects

Refresh R.	News	Shop	SoftCom	Cities	Mag
RMin	0,9329	0,9949	0,979	0,8421	0,9539
R10Min	0,8971	0,994	0,9786	0,8417	0,953
RHour	0,8379	0,9893	0,9765	0,8394	0,948
RDay	0,7519	0,8847	0,9196	0,7952	0,8899
RWeek	0,6439	0,4658	0,5632	0,6041	0,7716
RMonth	0,011	0,0017	0,0069	0,0000	0,1472

Table 10 The Cacheability Factor for HTML type objects

Refresh R.	News	Shop	SoftCom	Cities	Mag
RMin	0,2458	0,4966	0,2726	0,000	0,333
R10Min	0,2083	0,4662	0,2716	0,000	0,3296
RHour	0,000	0,318	0,2662	0,000	0,3111
RDay	0,000	0,000	0,1166	0,000	0,000
RWeek	0,000	0,000	0,0000	0,000	0,000
RMonth	0,000	0,000	0,0000	0,000	0,000

The HTML objects are nor fresh for longer then 1 day in any case. But even for those objects there is a notable difference between shop and cities websites. The CF for the former with refresh rate equal to 1 hour exceeds slightly 0.3 whereas it is equal to 0 for the latter. Judging by the character of the web sites the contrary should be the case.

7.3. Object duplication in Web sites

Websites have to preserve a distinct "look and feel" for a user. This is achieved among others by using the same graphical objects on many or even all pages of a given site. The scope of phenomenon is quite substantial as shown in the Table 11.

Table 11 Duplication of objects in test sequences

Site	Object no	Common objects	Total length	Common bytes	Site type
Londyn	44	34%	276267	21%	City
Wwa	46	63%	167414	45%	City
Wroc	111	57%	347171	44%	City
ChipPl	111	38%	296109	16%	Mag
ChipDe	195	48%	603490	39%	Mag
Linux	135	45%	171038	43%	Mag
CNN	161	36%	772209	28%	News
Rzepa	77	30%	236079	18%	News
Wp	126	37%	455577	17%	News
Aby	99	54%	388144	26%	Shop
Allegro	154	16%	481871	13%	Shop
Ebay	244	30%	937825	24%	Shop
Borland	106	50%	574997	35%	Soft
Prokom	92	62%	293362	50%	Soft
Softbank	119	41%	364718	47%	Soft

The duplication of objects could have a great impact on caching. In test sequences usually more then 1/3 of downloaded objects are loaded more then once. The test sequence is rather short in practice one could expect a for longer sequences witch would increase their scope of this phenomenon even further.

7.4. Cooperation of browser buffer and proxy server

A browser cache and a proxy server cache both operate according to similar rules of judging object freshness. A browser has a buffer large enough to keep all cacheable objects requested by a user in the recent several days. Only objects that are stale or not present al all in the local cache are requested from a proxy. The formulas presented in Section 4 refer only to such objects. A browser cache reduces their number. The Cacheability Factor introduced in the section 6.2 makes it possible to estimate the scope of this phenomenon. This time however one should analyze not the values of CF like it was done in this section but the changes in the values of CF for different refresh rates.

Let us assume that a proxy server is used by so many users that its refresh rate is R10min and a user visits the websites in test only once a day. These are plausible assumptions for a server with a large number of users visiting popular web sits such as these analyzed in the experiment. As a result some of objects that become stale in a browser cache of one user whereas they are fresh in a proxy cache due to their downloading by some other user. The scope of such objects in the initial phase of session could be estimated by the difference:

$$CF(x,R10Min)-CF(x,RDay).$$

The initial phase consists of the first loadings of a given page during a session. During the subsequent browsing a user revisits previously loaded pages. To estimate the

impact of proxy caches in this case one should take into account the refresh rates R10Min and RHour. The proxy advantage could be this time estimated by the difference:

CF(x,R10Min)-CF(x,RHour).

This should be treated as a maximum value as the usual refresh rate is usually somewhat less then 1 hour. The Table 12 shows the values for the tested website types.

Table 12 The scope of proxy caching

	Initial phase			Browsing phase		
	PRG	PIC	HTML	PRG	PIC	HTML
news	0,3767	0,1452	0,2083	0,0337	0,0592	0,2083
Shop	0,5865	0,1093	0,4662	0,0512	0,0047	0,1482
SoftCom	0,2222	0,059	0,155	0,0398	0,0021	0,0054
Cities	0,0396	0,0465	0,000	0,0014	0,0023	0,000
Mag	0,2246	0,0631	0,3296	0,0114	0,005	0,0185

The data presented in the Table 12 clearly indicate that even taking such a long value for browser refresh rate has not helped the proxies to influence the browser behavior in any significant manner. Proxies deliver significant amount of objects only in the initial phase of browsing. This is especially true for cache aware sites such as shops: the values in the table for the browsing phase are several times lover then for the initial phase. The cache negligence of the webmasters of the city web sites is visible here even more apparent then in the Tables 8 to 10.

8. Conclusions

Users demand that latency of loading internet pages should not exceed ca 8 seconds and the time span is largely not negotiable. The growing complexity and as a result the size of internet pages, the increasing volume of network traffic make it impossible to reduce the latency to acceptable levels by increasing the bandwidth of network connections alone. The buffering of data in proxy servers is widely considered to be option in that respect as it brings date closely to a user. The analysis presented in the paper shows that this assumption is not fully justified. The scope of cacheable date diminishes as more and more pages are dynamically created or personalized. The users have internet connections with significantly higher transfer rates then only few years ago and for many of them a proxy could not offer any advantage in the download speed. The browser caches have enough storage capacity and processing power to serve an individual user. Proxy servers have their place in the structure of Internet but it is because they increase the security of Internet access and reduce the volume of long distance traffic.

On the other hand a browser cache has unique opportunity to reduce the latency. It has all information about user preferences: what and when was requested by a user. The complicated problem of session identification encountered in Web usage mining therefore does not exist. Such an information is not be available on any other place in the Internet structure as users are concerned about their privacy. In addition to that the browser cache could compare the declared and actual change pattern of all of the requested objects. The web caches do not have neither the processing power nor the disk space to keep track of such detailed data. The transfer time from browser cache to browser is insignificant. A browser cache is not confined by the specifications of the HTTP protocol and can fully exploit the so called user dividend [17].

In order to reduce significantly the latency more flexible, augmented by object prefetchnig, algorithms should be developed.

9. Literature

1. Bhatti N., Bouch1 A., Kuchinsky A.: *"Integrating User-Perceived Quality into Web Server Design"*, Computer Networks (Amsterdam, Netherlands: 1999), vol. 33, no-16, pp.1-16
2. Brewington B.E., Cybenko G.: *"How dynamic is the {Web?}"*, Computer Networks; vol. 33,no: 1--6, pp 257--276, 2000
3. Bush V.: *"As we may think"*, The Atlantic Monthly, vol. 176 (1), pp.101-108, 1945
4. Card S.A., Moran T.P., Newell A.: *"The psychology of Human-computer Interaction"*, Lowrence Erlbaum Associates, NJ, 1983
5. Charny B." Who's the fattest site of them all?", ZDNet News, October 15, 2000,
6. Cohen R., Ramanathan S.: *"Using proxies to enhance TCP performance over hybrid fibber coaxial networks"*, Technical Report HPL-97-81, Hewlett-Packard Labs, 1997
7. Dikaiakos M.: *"Intermediary Infrastructures for the WWW"*, url: "citeseer.ist.psu.edu/dikaiakos02intermediary.html"
8. Feldman A,, Caceres R., Douglis F., Glass G., Rabinovich M.: *"Performance of Web proxy caching in heterogeneous bandwidth environments"*, in Proceedings of INFOCOM, pp 107-116, 1999
9. Gourley D., Totty B.: *"HTTP: the Definite Guide"*, O'Reilly & Associates, 2002
10. Jung J., Krishnamurthy B., Rabiniwich M.: *"Flash crowds and denial of service attacks Characterization and implications for CDN's and web sites"*, Proceedings of the International World Wide Web Conference, pp. 252--262, IEEE, May, 2002.
11. Kroeger T.., Long D., Mogul J.: *"Exploiting the bounds of Web latency reduction from caching and prefetching"*, in Proceedings od USENIX Symposium on Internet Technologies and Systems, pp. 13-22, 1997
12. Mobasher B., Cooley R., Srivastava J.: *"Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns"*, Journal of Knowledge and Information Systems, 1, 1999
13. Pearson M., McGregor A., Clearly J.: *"Reducing US/NZ Web Page Latencies"*, in Networks 99, Waikato University, Hamilton, New Zealand, 1999, pp. 54-63
14. Rabinovich M., Spatscheck O.: *"Web Caching and Replication"*, Addison-Wesley, 2002,
15. Ramsay J., Barbesi J., Preece J.: *"Psychological Investigation of Long Retrieval Times an the World Wide Web"*, Interacting with Computers, 10,1 (1998)
16. Sieminski A.: *"Buforowalnosc stron WWW"*, Multimedialne i sieciowe systemy informacyjne, MISSI 2004
17. Sieminski A.: *"The Potentials of Client Oriented Prefetching"*,, in Intelligent Technologies for Inconsistent Knowledge Processing, Advanced Knowledge International, 2004
18. Spero S.E.: *„Analysis of HTTP performance problems"* http://www.w3.org/Protocols/HTTP/1.0/HTTPPerformance.html
19. Valali A., Pallis G.: *"Content Delivery Networks: Status and Trends"*. IEEE Internet Computing 11, 2003 pp. 68-74
20. Wessels D.: „Can WWW caches help save the Internet?", http://www.caida.org/outreach/papers/1996/wwwcaches/wwwcaches.html
21. Wessels D.:" *Web Caching*", O'Reilly and Associates, Inc,. 2001
22. Zhang X.:"*Cachability of Web Objects*", Technical Raport 2000-019, Boston U., CS Department , 2000, citeseer.ist.psu.edu/zhang00cachability.html
23. Zona Research: *„The economic impacts of unaccaptable Web Site download speeds"*, White paper, http://www.zonaresearch.com/deliverables/while-

papers/wp17/index.html

WWW Resources:

W1 ftp://ircache.nlanr.net/Traces
W2 http://www.akamai.com/
W3 http://www.faqs.org/rfcs/rfc2616.html
W4 http://www.ircache.net/cgi-bin/cacheability.py
W5 http://www.numion.com/
W6 http://www.squid-cache.org/