Efficiency of Caches for Content Distribution on the Internet

Gerhard Haßlinger
Deutsche Telekom Netzproduktion
Darmstadt, Germany
gerhard.hasslinger@telekom.de

Oliver Hohlfeld
Technische Universität Berlin / DT Labs
Berlin, Germany
oliver@net.t-labs.tu-berlin.de

Abstract — **Traffic engineering and an economical provisioning of bandwidth is crucial for network providers in times of high competition in broadband access networks. We investigate the efficiency of caching as an option to shorten end-to-end paths and delays while at the same time reducing traffic loads. The portion of HTTP based distribution of cacheable content on the Internet is increasing in recent time. In addition, the favourable effect of Zipf-like access pattern on caches is also confirmed for currently most popular web sites with user generated content. Content delivery (CDN) and peer-to-peer (P2P) networks are distributing a major portion of IP traffic with different impact on caching. P2P traffic is subject to long transport paths although appropriate for caching in principle. CDNs are based on server infrastructures allowing for shorter paths on a global scale on top of network provider platforms.**

We give a brief overview of the options for deploying caches by content and network providers at different points in the interconnection, backbone or aggregation. The main part of the work focuses on the analysis of replacement strategies with regard to Zipf-like and fixed or slowly varying access pattern. A comparative evaluation shows that least recently used (LRU) essentially differs from caching strategies based on access statistics in terms of the achievable hit rates.

***Index Terms* — Content delivery, P2P, CDN overlays, caches, Zipf distributed access pattern, transport path optimization.**

I. CACHING FOR GROWING INTERNET TRAFFIC

TRAFFIC on networks for broadband Internet access is steadily increasing although at different pace over time depending on the deployment stages of technology in different regions. The Minnesota Internet Traffic Studies (MINTS) [26] give an overview including links to many relevant sources with measurement data from traffic exchange points and reference to official statistics of some countries, e.g., for Australia or Hong Kong [4][30]. Based on that, Odlyzko et al. [29] estimated a 100% traffic growth rate per year in core areas of the Internet on the average from 1990 to 2002. Meanwhile the global annual traffic growth slowed down to about 45% [26] as confirmed also in a white papers series on IP traffic by Cisco Systems [11]. Main current trends and forecasts are:

- Global IP traffic will nearly double every two years through 2013.
- Peer-to-peer (P2P, file sharing etc.) is growing in volume, but declining as a percentage.
- Video is a major source of IP traffic growth. All types of video will account for >90% of consumer traffic in 2013.

There is a discrepancy between the access speed provided in broadband access networks and the available bandwidths in core networks. According to periodically updated statistics by the content delivery network provider Akamai [3], several Asian countries provide the highest speeds for broadband access per users (South Korea: 12Mb/s, Hong Kong: 8Mb/s, Japan, Romania: 7Mb/s,) followed by countries in Europe and North America in the range of 3-6Mb/s. In Germany the access speeds of well beyond 15 million broadband access lines with mean rate of 4Mb/s [3] sum up to a total access capacity of >50Tb/s for private households. If most users would exploit their access speed then even 100Gb/s IP and Ethernet links as currently standardized for next generation backbone equipment would be insufficient, while a next wave of video and IP-TV applications up to HDTV quality [14] is expected to further raise traffic demands.

In this situation, traffic engineering measures for load balancing and optimizing traffic paths will stay important. We investigate options for and the efficiency of caching. Caches have been deployed on nodes in the Internet since more than a decade in order to shorten the transport paths for a considerable portion of requests referring to popular web sites. Several studies have shown the efficiency of caching around the millennium [5][26]. Then P2P networking contributed the major portion of Internet traffic for several years, which bypassed classical HTTP based web caches and made them inefficient. In recent time, user generated content becomes even more relevant, but server based platforms like YouTube and social networks have grown faster than data exchange via P2P, such that HTTP requests currently again account for most of the IP traffic. As a consequence, the portion of cacheable content is increasing [16] where extremely skewed Zipf-like access pattern are again observed in recent studies on YouTube and other platforms [7][14][16][17]. Besides network providers, the operators of content platforms often make use of globally distributed content delivery networks [3][34] with similar effect on shortening transport paths.

This paper studies the deployment of caches by content and network providers at different points in the interconnection, backbone and aggregation level. The main part of this study is devoted to cache replacement strategies, where fundamental research already started several decades ago on the same problem for paging in operating systems [15][25]. When access pattern are stable over a considered time span then an optimum strategy is to put the most popular items in the cache, where small caches are already expected to be efficient for Zipf-like access distributions [8][10][14][16]. The exact hit rate analysis for usual strategies can be solved in closed form [15][25][33] but involves evaluation of combinations of items in the cache, which render infeasible already for moderate cache sizes. Instead, we propose simpler modeling of strategies in an approximate but scalable analysis. A similar approach has been evaluated for LRU versus FIFO caching for database applications [13]. Compared to [13] this paper provides an update for web caches with different conclusions for Zipf distributed requests and adds a comparison to replacement strategies based on statistics about requests over a recent time frame.

The next section discusses transport paths and delays in CDN and P2P networks as overlays with different impact on caching. Section III briefly addresses options for caching within the architecture of network providers followed by an approximate performance analysis for replacement strategies using access statistics and LRU in Sections IV and V. Sections VI and VII show comparative evaluation results in terms of the hit rates for caching strategies with special focus on Zipf-like access pattern and available traces. Concluding remarks and an outlook are given in section VIII.

II. CDN and P2P Paths for Content Delivery

The end-to-end transport paths naturally depend on the applications with fundamental differences between client-server based and peer-to-peer networking. A large portion of Internet traffic is currently delivered via CDN and P2P overlay networks [3][11][18][21][27]. Content delivery (or distribution) networks are based on server farms within the Internet infrastructure, whereas peer-to-peer networks are organized on the terminal equipment of the users.

A. Content delivery networks

CDNs are developing from support of popular web sites to include streaming, IP-TV and many other Internet services. A study of transfer paths through Akamai´s CDN [34] shows how users are redirected from the main web site via an hierarchical server farm to one of more than 10 000 CDN servers, which has a short path to the destination. The connection is dynamically switched between servers if performance measurement or server load indicate better quality of service on an alternative backup path. Su et al. [34] confirm that CDNs are efficient in shortening transport paths and improve delays and throughput as main quality of service characteristics.

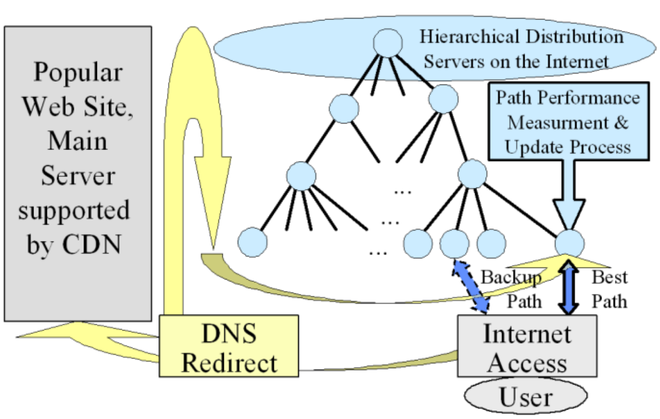

Figure 1: Content delivery networks

B. P2P networks

P2P networks are known to be highly efficient and scalable in distributing large data volumes among big user communities by exploiting storage, computational power and access bandwidth as otherwise vacant resources of the users. P2P overlays involve only a minimum of network infrastructure and but currently lead to long transport paths spanning around the globe between randomly chosen peers. Since popular content is found in many replicas in a P2P network, sources for download are often available close to a requesting user. Therefore a preference for local downloads can essentially shorten end-to-end paths and reduce the load on backbone and expensive interconnection links [2]. Since 2008 solutions to get information on node locations within the Internet connectivity are addressed by an IETF working group [24] in order to prefer close-by sources in distributed applications.

C. Comparison of CDN and P2P paths and delays

As compared to CDN transfers from a close-by server, P2P downloads usually experience much longer paths and delays which also affect throughput and reliability. For network providers, unnecessary long transfer paths impose higher load on peering and interconnection routes including expensive intercontinental links [21][24].

Considering large provider networks serving millions of subscribers, it can be expected that a majority of the data of a global file sharing network is already found to be replicated on the same ISP platform and partly in the same access region of a P2P downloader. The fact that the major portion of downloads is addressed to a small set of the currently most popular files strengthens this effect. Sometimes a tendency for local P2P content exchange arises within social groups. A separation of user communities and content due to different languages is most obvious [32].

We have evaluated the delays for traffic via P2P and CDN overlays through packet based measurement on links in the aggregation of Deutsche Telekom's broadband access network [21]. We did not try to capture all traffic of both types, but selected a fraction that can be easily detected via P2P ports for BitTorrent, eDonkey and Gnutella and via IP address ranges indicating autonomous systems of Akamai, Limelight, Google and other server sites. The flows classified via P2P ports made a fraction of 2.7% of the traffic volume, while the IP address ranges for CDNs and popular web sites accounted for 10.7% of the total traffic.

A packet measurement on two 1Gb/s links in parallel in the aggregation in downstream direction yielded a total mean traffic rate of about 820 Mb/s during a time span of one hour at the daily peak time in mid 2008. We used two successive packets sent by the client in the TCP handshake to estimate the round trip time (RTT), although this may also include response times of the server or peer in response to the TCP connection request as well as delay jitter on the path to the measurement point.

The mean RTT delays evaluated in Table 1 are 2.5-fold higher for P2P, which is also apparent when looking at delays beyond bounds, where almost 10% of the P2P flows have RTT longer than 1s. Such delays would be unacceptable for real time applications.

Comparison of RTT Statistics	Mean	RTT > 0.1s	RTT > 0.2s	RTT > 0.5s	RTT > 1s
CDN	0.125s	30.9%	11.9%	4.2%	0.9%
P2P	0.330s	74.1%	43.2%	20.1%	9.5%

Table 1: Delay measurement for P2P and CDN overlays

III. Caches in Broadband Access Networks

The business case for caching is determined in a tradeoff between cost savings for capacity on transmission links and costs for the caches. The bandwidth saving due to caching basically depends on the frequency of accesses to cacheable content.

Figure 2 shows the architecture of broadband access networks including connectivity to content platforms via the IP backbone and aggregation network. Network providers have options to install caches on nodes in the aggregation network, in the backbone or at peering links. In addition, content delivery networks provide caching systems on a global scale and caching is also applied in the user's end systems, see [5] for an overview of user and content oriented caching methods.

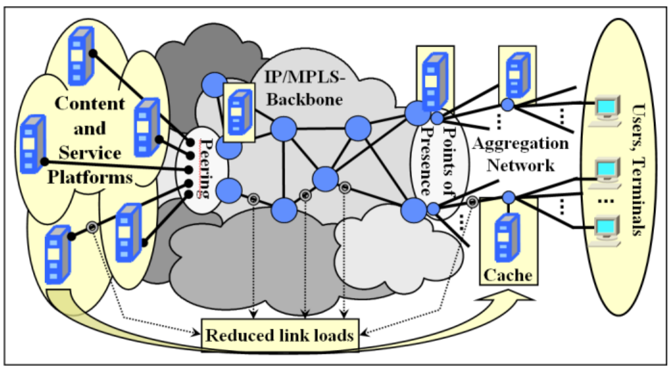

Figure 2: Caches in broadband access networks

Caches close to the users can cut off most of the transmission path. On lower aggregation levels many caches have to be deployed, each of whom is serving a smaller user population with the consequence of smaller hit rates. While the access pattern of users is not expected to vary much in different regions of broadband access for homes, the access pattern through corporate networks or in organizations can essentially differ from the global access behavior. The study [16] experienced such differences in the popularity of YouTube content accessed from a university. This can lead to higher efficiency for local caches adapted to user communities. In general, the efficiency of a cache with storage for a set $I = \{o_1, o_2, \ldots, o_M\}$ of items depends on

- a_k: the request frequencies of items o_k in the cache e.g. in terms of the expected number of requests per day,
- s_k : the size of the items o_k
- $c_{Cache}(s_k)$: costs for caching the items o_k
- c_{Lj}^{BW}: costs for bandwidth provisioning on links L_j in the broadband access network.

It is worthwhile to use caching and to store an item o_k in the cache, if the costs are lower than the positive effect of shortened transmission paths. In particular, let $L_1, L_2, \ldots, L_S$ denote the links on the transmission path that are bypassed when using a cache. A precondition for cost efficient caching is given by

$$c_{Cache}(s_k) < \sum_{j=1}^{S} a_k s_k c_{L_j}^{BW}, \qquad (1)$$

where $a_k s_k$ represents the bandwidth saved by caching o_k. In addition to the size of the item and the traffic driven costs of links in the network, there are costs associated with operations and maintenance of the links as well as the caches not considered in (1), which require much more elaborate weighting. A network provider's decision on whether and where to install caches has to be based on a business case study covering all options in a network wide architectural view, involving access pattern for relevant service and content platforms. The balance of costs (1) may be used to check for appropriate positioning of caches and to decide where to put an item if caches are available at different nodes in a transport path.

In principle, cacheability is improving on the Internet in recent time [1][16]. P2P network traffic is partly shifting towards transport via the HTTP protocol. Although P2P traffic is appropriate for caching [6][31], P2P caches do not seem to be deployed and P2P caching options introduced e.g. by the eDonkey protocol are again embedded into HTTP [32].

A main problem of cache efficiency as with quality of service (QoS) support in general are administrative boundaries of content and network providers, where limited cooperation may detract from optimization over the complete end-to-end transfer path. CDN support for large content providers often ends on peering links at the boundaries of network providers who engineer the traffic within their platform more or less independently. In order to make full use of caching, unique hash identifiers are proposed for each content item to detect and avoid duplicate transport over the same path [28], as also successfully applied in P2P networks.

However, up to now no such solution seems to be standardized and supported by major content providers. Thus detecting that different requests address the same content in the cache is not trivial and can add considerable performance burden on caching systems, even if Web 2.0 provides helpful meta-data to improve the efficiency of caching [1][16]. On the other hand, content platforms like YouTube often personalize the handling of requests in a way that makes it even more difficult to classify requests addressing the same item based on HTTP data, although both the content and network provider would benefit from keeping large popular content volumes in caches close to the user. Further problems of outdated data in a cache may be less relevant for video streaming or downloads which drive the Internet traffic growth but are still important for access to web sites with dynamically updated information.

IV. Hit Rate Analysis for LRU Caching

Let $I = \{o_1, o_2 \ldots o_N\}$ denote a set of cacheable items. We assume that requests are identical and independently distributed (i.i.d.) within the set I in a considered time frame such that each request refers to an item o_k with probability a_k without memory of previous accesses. The items o_k are ordered according to decreasing access probabilities $a_1 \geq a_2 \geq \ldots \geq a_N$. Access statistics are measured on many web sites for commercial reasons and partly shown e.g. by YouTube, Amazon etc. The operator of a web server has statistics on recent trends available, which are expected to provide useful approximations for access probabilities also in the near future.

We assume that the cache C_M has limited storage for $M\,(< N)$ arbitrary objects irrespective of the item size. Usually the item sizes are relatively small as compared to the size of the cache although cache sizes are again small compared to the content on web platforms that may be cached. Then the variability in item sizes does not have a major impact on the cache hit rate. In addition, many Internet applications divide larger content volumes into small segments that are better suited for real time transmission and reduce the variability of the size of cacheable objects. For independent accesses the maximum hit rate $R_{C_M}^{\max}$ is achieved when the most popular objects are in the cache:

$$R_{C_M}^{\max} = a_1 + a_2 + \ldots + a_M . \qquad (2)$$

Provided that the access pattern of a web server or another caching environment is static or slowly varying according to independent access probabilities, then a near optimum caching strategy is to estimate the probabilities from access statistics over an appropriate time frame and to keep the most often referenced items in the cache as considered in section V. On the other hand, access statistics for web applications are more or less dynamic up to extreme effects of spontaneously arising flash crowds interested in some newly appearing content. Therefore the content of caches has to be steadily adapted based on a replacement strategy to include new promising items and to select others to be removed.

Least recently used (LRU) and *first in first out* (FIFO) are simple and usual replacement principles. LRU is realized by ordering the items in the cache such that a newly referenced item is always put on top while previously higher-ranked items step down by one rank. With FIFO strategy, a new top item and reordering is only initiated if an item outside of the cache has been referenced. As a consequence, LRU keeps more often accessed items longer in the cache and comes closer to a strict preference according to access statistics, as confirmed in all evaluations in [13] for comparison of LRU and FIFO. Therefore we omit FIFO in our considerations.

We analyze the hit rate for LRU in order to compare it to the optimum strategy of having the most popular items in the cache. When we consider a cache C_M of size M then the last M references to different items determine the current content in the LRU cache. Starting with $M = 1$, the last access determines the top item such that the probability to find o_k in the cache C_1 is equal to the access probability $p(o_k \in C_1) = a_k$. Thus the hit rate R_{C1}^{LRU} of a LRU cache for one item is

$$R_{C_1}^{\mathrm{LRU}} = \sum_{k=1}^{N} p(o_k \in C_1)\, a_k = \sum_{k=1}^{N} a_k^2 . \qquad (3)$$

The probability that the LRU cache contains o_k on top and o_j in the second rank is given by a_k for the last access combined and thus multiplied with $a_j / (1 - a_k)$ for the previous access to an item $o_j \neq o_k$. Therefore the hit rate for a 2-item cache is

$$R_{C_2}^{\mathrm{LRU}} = \sum\nolimits_{k=1}^{N} a_k \sum\nolimits_{j=1, j\neq k}^{N} \frac{a_j}{1-a_k} (a_k + a_j). \qquad (4)$$

The analysis extends to caches of arbitrary size $M < N$ [33]:

$$R_{C_M}^{\mathrm{LRU}} = \sum_{k_1=1}^{N} a_{k_1} \sum_{\substack{k_2=1\\ k_2\neq k_1}}^{N} \frac{a_{k_2}}{1-a_{k_1}} \sum_{\substack{k_3=1\\ k_3\neq k_1,k_2}}^{N} \frac{a_{k_3}}{1-a_{k_1}-a_{k_2}} \ldots \sum_{\substack{k_M=1\\ k_M\neq k_1,\ldots,k_{M-1}}}^{N} \frac{a_{k_M}}{1-\sum_{j=1}^{M-1} a_{k_j}} \sum_{j=1}^{M} a_{k_j} . \qquad (5)$$

The sum involves all combinations of M items which render the evaluation infeasible for large M. Instead we propose an approximation of $R_{C^M}^{\mathrm{LRU}}$ with moderate computational complexity for large caches. Therefore we consider the probabilities $p(o_k \in C_M)$ that an item o_k is found in a cache of size M, and that it is especially found on rank r in the previously described LRU order. On top we again have $p(o_k \in C_{M-1}) = a_k$ for the item addressed in the last request. The item o_k is found in the second rank when the last access referred to an item $o_j \neq o_k$, i.e. with probability $a_j / (1 - a_k)$ and the previous access selects o_k among all elements different from o_j i.e. with probability $a_k/(1 - a_j)$:

$$p(o_k \in C_2) = p(o_k \in C_1) + (1 - p(o_k \in C_1)) \sum_{\substack{j=1\\ j\neq k}}^{N} \frac{a_j}{1-a_k} \frac{a_k}{1-a_j} \qquad (6)$$
$$= a_k \left(\frac{1-2a_k}{1-a_k} + \sum_{j=1}^{N} \frac{a_j}{1-a_j} \right).$$

While this formula is exact and corresponds to equation (4) for R_2^{LRU}, we propose a less complex approximation for $M > 2$.
We compute the probability of finding item o_k in position M of the LRU stack

$$p(o_k \in C_M) - p(o_k \in C_{M-1}) = (1 - p(o_k \in C_{M-1})) \frac{a_k}{1 - R_{C_{M-1}/\{o_k\}}^{\mathrm{LRU}}} \qquad (7)$$

considering the situation when o_k has been selected at the M-th backward stage among all items not in C_{M-1}. The denominator provides normalization over the access probabilities of items not in the cache, where the hit rate of a cache of size M–1 is subtracted, which is known to exclude o_k. The computation is completed by determining the latter hit rates by

$$R_{C_{M-1}/\{o_k\}}^{\mathrm{LRU}} = \sum_{\substack{j=1\\ j\neq k}}^{N} p(o_j \in C_{M-1})\, a_j \frac{M-1}{M-1-p(o_k \in C_{M-1})}. \qquad (8)$$

Since $\sum_j p(o_j \in C_{M-1}) = M-1$, a corresponding weighting factor accounts for the exclusion of o_k.

Equations (7 - 8) provide an iterative scheme to compute $p(o_j \in C_M)$ for $M = 2, 3, 4, \ldots$ with initialization for C_2 in equation (6). The scheme avoids considering all possible combinations of items in C_M and approximately inserts the conditional hit rates (8). The computational complexity of the complete analysis is $O(NM)$ to evaluate (7 - 8) up to C_M and stays tractable up to millions of items. Finally we obtain the hit rate

$$R_{C_M}^{\mathrm{LRU}} = \sum\nolimits_{j=1}^{N} p(o_j \in C_M)\, a_j .$$

The presented framework recovers results by Dan and Towsley [13], where especially equations (1) and (2) in [13] correspond to the previously equations (7) and (8), when we apply the finest granularity to identify each partition in [13] with an item. Then equation (8) is replaced in [13] by the simpler term

$$R_{C_{M-1}}^{\mathrm{LRU}} = \sum\nolimits_{j=1}^{N} p(o_j \in C_{M-1})\, a_j$$

i.e. the computation of hit rates during the iteration does not regard the fact that item o_k is not in the cache when evaluating (7). The difference is negligible for small access probabilities a_k, but significant for popular items with e.g. $a_k > 0.5$.

Dan and Towsley do not consider Zipf distributed accesses as observed in the Internet, but focus in their analysis on truncated arithmetic and truncated geometric distributions for database applications. Although they partly observe that LRU provides a close to optimal performance, our evaluations of LRU with Zipf distributed accesses in Section VII always shows significant LRU performance degradations.

Next, we transfer the approximate analysis to replacement methods based on access statistics.

V. HIT RATE ANALYSIS FOR REPLACEMENT BASED ON ACCESS STATISTICS WITH LIMITED BACKLOG

Next we use statistics over a sliding window of the last L accesses to prefer the most often referenced items for caching. Again we assume independent and identically distributed (i.i.d.) references to items with probabilities $a_1 \geq a_2 \geq \ldots \geq a_N$. When the statistics has only a short memory of a few recent accesses then we expect essential random deviations, but with increasing L a tendency to a ranking according to the access probabilities is expected to prevail. We are interested in finding out which backlog L is required in order to give sufficient preference for the top items $o_1, o_2 \ldots o_M$ in the cache in order to approach the optimum hit rate $R_{C_M}^{\max} = \sum_{k=1}^{M} a_k$.

Let $\#_k$ denote the number of references to o_k within the last L accesses. Then the probabilities that m out of L accesses are addressed to o_k follow a binomial distribution

$$p(\#_k = m) = \binom{L}{m} a_k^m (1-a_k)^{L-m}.$$

The distribution can be equivalently expressed using generating functions such that

$$G_{\#_k}(z) \overset{def}{=} \sum_{m=0}^{L} p(\#_k = m)\, z^L = ((1-a_k) + a_k z)^L$$

Next we determine the probabilities $p(\#_j = n \mid \#_k = m)$ for pair wise comparison of the number of accesses to items which determines their order in the ranking:

$$p(\#_j = n \mid \#_k = m) = \binom{L-m}{n} \left(\frac{a_j}{1-a_k}\right)^n \left(1 - \frac{a_j}{1-a_k}\right)^{L-m-n}.$$

The rank r_j of o_j is smaller if the number of accesses is larger $\#_j > \#_k \Rightarrow r_j < r_k \Rightarrow p(r_j < r_k | \#_k = m) = \sum_{n>m} p(\#_j = n \mid \#_k = m)$ An exact determination of the rank r_k of o_k based on statistics of the last L accesses includes probabilities for all combinations of the number of accesses to the entire set of items. We simplify the computation of rank probabilities $p(r_k = l)$ by assuming that pair wise rankings are independent of each other, i.e. that $p(\#_j = n \mid \#_k = m)$ is independent of $p(\#_i = n \mid \#_k = m)$ for $i \neq j$ and as a consequence

$$p((r_i < r_k) \wedge (r_j < r_k) \mid \#_k = m) = p(r_i < r_k \mid \#_k = m)\, p(r_j < r_k \mid \#_k = m).$$

Then the generating function for the distribution of the rank r_k can be expressed as

$$G_{(r_k \mid \#_k = m)}(z) = \sum_{l=1}^{N} p(r_k = l \mid \#_k = m)\, z^l$$
$$= z \prod_{\substack{j=1 \\ j \neq k}}^{N} \left(1 - p(r_j < r_k \mid \#_k = m) + p(r_j < r_k \mid \#_k = m)\, z\right)$$

where the factor z accounts for the fact that $r_k \geq 1$ and each factor of the product represents a 0-1-distribution with regard to the comparison of r_j and r_k.

The resulting probabilities $p(r_k = l \mid \#_k = m)$ indicate whether o_k can be found in the cache C_M. o_k is clearly outside the cache if $r_k > M$, but for $r_k \leq M$ multiple items in the same rank $\#_j = \#_k$ have to be taken into account. Let $N(r_k)$ denote the number of items in equal rank with o_k. If $r_k + N(r_k) - 1 \leq M$ then o_k and all items in the same rank fit into the cache. Otherwise, if $r_k \leq M$ and $r_k + N(r_k) - 1 > M$ then only $M - r_k + 1$ of those items can be placed in the cache. We assume random selection, such that the probability for o_k to enter the cache is $(M - r_k + 1) / N(r_k)$ in such cases.

The generating function and thus the probability distribution of $N(r_k)$ is computed similar to the rank distribution by considering $p(r_j = r_k \mid \#_k = m)$ instead of $p(r_j < r_k \mid \#_k = m)$:

$$G_{(N(r_k) \mid \#_k = m)}(z) = \sum_{l=1}^{N} p(N(r_k) = l \mid \#_k = m)\, z^l$$
$$= z \prod_{\substack{j=1 \\ j \neq k}}^{N} \left(1 - p(\#_j = m \mid \#_k = m) + p(\#_j = m \mid \#_k = m)\, z\right)$$

Once the probabilities of the rank distribution and of the number of equally ranked items are determined, the probability of finding o_k in the cache is given by

$$p(o_k \in C_M \mid \#_k = m) = \sum_{j=1}^{M} p(r_k = j \mid \#_k = m) \cdot$$
$$\left(\sum_{l=1}^{M-j+1} p(N(r_k) = l \mid \#_k = m) + \sum_{l=M-j+2}^{N} p(N(r_k) = l \mid \#_k = m)\, \frac{M-j+1}{l}\right).$$

From the conditional rank distributions for $\#_k = m$ we finally obtain the probability for o_k being in the cache for arbitrary $\#_k$

$$p(o_k \in C_M) = \sum_{m=0}^{L} p(\#_k = m)\, p(o_k \in C_M \mid \#_k = m)$$

and the cache hit rate $R_{C_M} = \sum_{k=1}^{N} p(o_k \in C_M)\, a_k$.

The computational complexity of the evaluation is

- $O(NL)$ for the binomial distributions $p(\#_k = m)$ for all items m,
- $O(N^2L^2)$ for $p(\#_j = n \mid \#_k = m)$ for all pairs of items,
- $O(NM^2L)$ for $G_{(r_k \mid \#_k = m)}(z) \Leftrightarrow p(r_k = l \mid \#_k = m)$ for ranks $r_k < M$ as required to check if o_k is in the cache,
- $O(N^3L)$ for $G_{(N(r_k) \mid \#_k = m)}(z) \Leftrightarrow p(\#_j = m \mid \#_k = m)$,
- $O(N\,L)$ for $p(o_k \in C_M \mid \#_k = m)$ and the final computation of the cache hit rate.

In total, the computational effort is of the order $O(N^2L(N+L))$. Most probabilities $p(\#_k = m)$, $p(\#_j = n \mid \#_k = m)$ are negligibly small for large N, L. Thus a restriction on non-negligible ranges essentially reduces the computation effort.

The computation is subject to inaccuracies introduced by the assumption that pair wise rankings are independent of each other in determining rank distributions $p(r_k = l \mid \#_k = m)$. Such deviations are most relevant for small L. For $L = 1$ only the last access is included in the access statistics such that item o_k is put in the first rank with probability a_k and all items not on top are equally ranked. Then we obtain as an exact result

$$p(o_k \in C_M) = a_k + (1 - a_k)(M-1)/(N-1);$$
$$R_{C_M} = \sum_{k=1}^{N} \left(a_k + (1-a_k)\frac{M-1}{N-1} \right) a_k = \frac{M - 1 + (N-M)\sum_{k=1}^{N} a_k^2}{N-1}$$

This result corresponds to fixed values, e.g. $p(r_k = 2 \mid \#_k = 0) = 1$ for the rank and $p(N(r_k) = N-1 \mid \#_k = 0) = 1$ for the number of equally ranked items, whereas the previously proposed computation scheme leads to Poisson-like distributions in both cases with the same mean but with a non-zero variance. The deviations are essential for most small M.

In general, the preference for items according to statistics of L recent accesses shows monotonously increasing hit rates, starting from random caching of items corresponding to $L = 0$ with hit rate M/N and approaching $\sum_{k=1}^{M} a_k$ for large L.

The binomial distributions $p(\#_k = m)$ with mean $L\, a_k$ and standard deviation $\sqrt{L a_k (1-a_k)}$ also approaches a Poisson distributions with for large L. For each pair of items with $a_j < a_k$ the differentiation of their rank will become stronger for increasing L since the standard deviations are increasing slower than the mean such that the differences in the mean are becoming more significant until finally

$$a_j < a_k \quad \Rightarrow \quad lim_{L \to \infty}\ p(\#_j \geq \#_k) = 0.$$

The evaluation section VII will show this development for some examples and estimate the window size L for the statistics in order to get close to the optimum hit rate.

VI. ZIPF LAWS FOR ACCESS TO POPULAR CONTENT

Pareto and Zipf distributions are observed manifold in Internet statistics or more general, when a large population is accessing a large set of items. According to a Zipf law, the item in rank R in the order of highest access frequency attracts

$$A(R) = \alpha R^{-\beta} \qquad (\alpha > 0;\ \beta > 0) \tag{9}$$

requests. The parameter $\alpha = A(1)$ can refer to the maximum number of requests observed for an item in the statistics or is determined by a normalization constraint $\Sigma_R\ A(R) = 1 \Rightarrow \alpha = 1/\Sigma_R R^{-\beta}$ such that $A(R)$ expresses the portion of accesses in a probability distribution. The exponent β determines the differences in access frequencies and thus the variance of the distribution, which is increasing with β.

The relevance of Zipf distributions has been confirmed manifold in the Internet, e.g.,

- for popular web sites by Breslau et al. [7] in a series of measurements yielding $0.64 < \beta < 0.85$ as exponents,
- for access frequencies in distribution platforms like YouTube [8][10][16] with $\beta \to 1$ and many other content and video platforms referenced in [16], America Free TV <americafree.tv> [14], Amazon <www.amazon.com>, P2P networks [6] etc.
- for further examples mentioned e.g. by M. Eubanks [14] from cross references in the Internet to relationships in large social networks or for the frequency of words in a long text.

Zipf laws lead to bounds on the sum of request frequencies of the top N items:

$$\sum_{R=1}^{N} R^{-\beta} > \int_{R=1}^{N+1} R^{-\beta} dR = \left. \frac{R^{1-\beta}}{1-\beta} \right|_{1}^{N+1} = \frac{(N+1)^{1-\beta} - 1}{1-\beta} \tag{10}$$

where the sum corresponds to an integral of a step function $f_U(R) = \lfloor R \rfloor^{-\beta} \geq R^{-\beta}$ as an upper bound of the real valued function $R^{-\beta}$. Vice versa, we obtain the lower bound

$$\begin{gathered} f_L(R) = \lfloor R+1 \rfloor^{-\beta} < R^{-\beta} \Rightarrow \\ \sum_{R=1}^{N} R^{-\beta} = 1 + \sum_{R=1}^{N-1} (R+1)^{-\beta} < 1 + \int_{R=2}^{N+1} R^{-\beta} dR < \frac{(N+1)^{1-\beta} - \beta}{1-\beta}. \end{gathered} \tag{11}$$

When a cache of size for M items is available for a larger set of $N > M$ items, then equations (10-11) also bound the fraction of accesses to the top M items within the set of N items:

$$\frac{(M+1)^{1-\beta} - 1}{(N+1)^{1-\beta} - \beta} < \frac{\sum_{R=1}^{M} R^{-\beta}}{\sum_{R=1}^{N} R^{-\beta}} < \frac{(M+1)^{1-\beta} - \beta}{(N+1)^{1-\beta} - 1} \tag{12}$$

Figures 3 and 4 show the shape of Zipf distributions with emphasis

- on the influence of the exponent β in Figure 3 and
- on the size M of the set of items in Figure 4.

The curves confirm the influence of a small set of top items, which attract a considerable amount of all accesses. When the exponent β is increasing from 0.5 to 1 then the fraction of accesses to top elements is growing by 10% - 30% with most effect for the top 1%.

While Zipf distributions always tend to become more skewed for larger β, the influence of the size M of the set of items is less straightforward. When a fixed number of top M items is considered then the corresponding fraction of accesses is decreasing with the size N of the set of items:

$$\lim_{N \to \infty} \frac{\sum_{R=1}^{M} R^{-\beta}}{\sum_{R=1}^{N} R^{-\beta}} < \frac{(M+1)^{1-\beta} - \beta}{(N+1)^{1-\beta} - 1} \to 0. \tag{13}$$

When the top K% items are considered such that $M = NK/100$, then the fraction of accesses to those top items is increasing with the set of items N. Figure 4 shows a slight increase for small exponents in three curves for $\beta = 0.6$ and a larger increase for high exponents in the curves for $\beta = 1$.

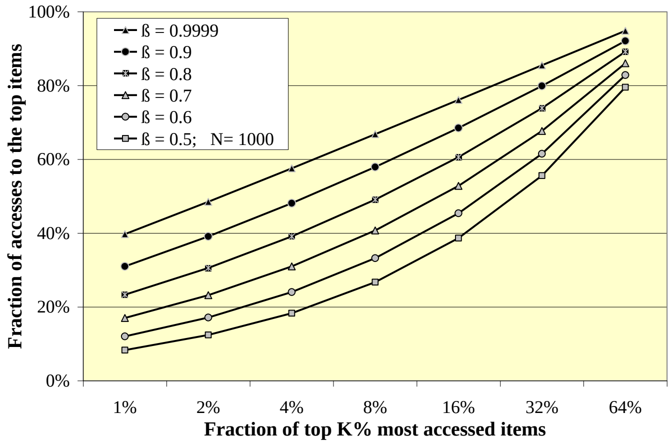

Figure 3: Skewness of Zipf distributed accesses $A(R) = \alpha R^{-\beta}$: Impact of the exponent β

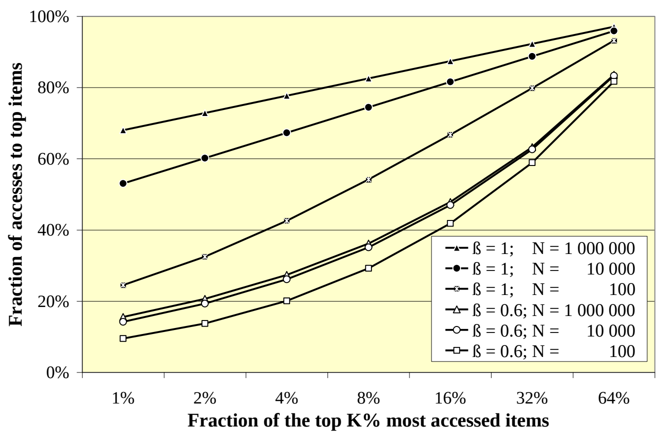

Figure 4: Skewness for Zipf distributed accesses: Impact of the size N of the set of items

Zipf laws typically yield a good match only for a limited range with deviations often being observed for the most popular and for seldom accessed items [7][14][16]. Alternatively, the study [17] e.g. proposes the term

$$A(R) = (b - a\ ln(R))^{c} \quad (14)$$

with three parameters for adaptation in cases with less skewed access distributions than for Zipf laws.

VII. Evaluation of Cache Hit Rates

We apply the analysis for the caching strategies of sections IV and V to cases of Zipf distributed access and to subsets of YouTube access traces provided by [8]. The first example in Figure 5 compares the approximate LRU analysis to a simulation and to static allocation of the most popular items as the best case. Access probabilities are determined based on statistics for the first 50 000 items in the YouTube trace [8]. The results show the skewness of accesses, where the top 256 items 50% hit rate when stored in a cache of this size. LRU hit rates are 10-20% below the optimum case and simulation results show an almost perfect match of the approximate analysis.

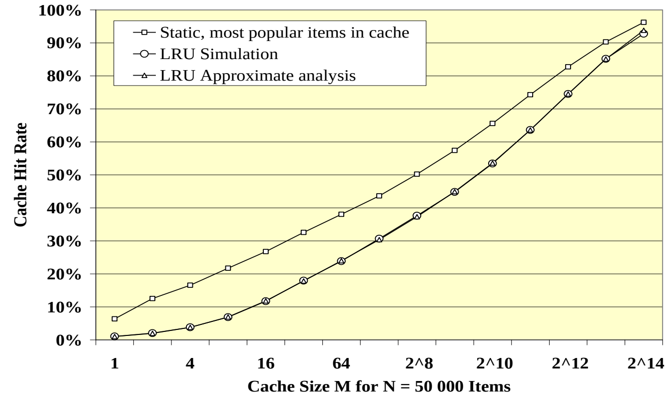

Figure 5: Performance of LRU caching

The example in Figure 6 sets access probabilities according to statistics for the first 1000 items in the YouTube trace [8]. We compare the hit rate for the strategies of putting the most popular items in the cache, for LRU and for access statistics as studied in Section V over a sliding window of L = 200 with cache size M = 50. Figure 6 shows that LRU again remains 10-20% below the optimum, which is inefficient especially for small caches. The access statistics is close to the optimum for small caches, since it prefers the top items, but becomes worse than LRU for large caches. This indicates that a larger window size L > 200 is required for better statistical differentiation of items to be put into the cache.

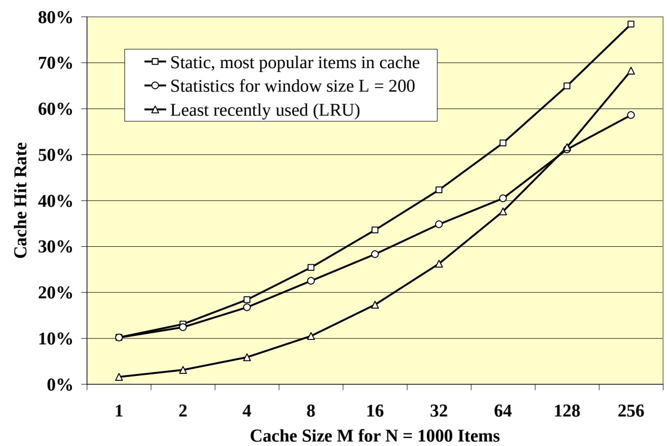

Figure 6: Comparison of cache hit rates

Another example investigates the effect of the window size L for including backward accesses in the statistics approach. Starting with the memoryless case L = 0 with randomly cached items the probability of having the popular items in the cache is steadily increasing as shown in Figure 7. This also holds for the cache hit rate (L = 0: 10%, L = 4: 20.3%, L = 64: 50.8%, L = 1024: 63.8%) which approaches the optimum of 66.3% for static allocation.

VIII. Conclusion and Outlook

The positioning and distribution of content on the Internet in order to achieve short transport paths to each requesting user has a major impact on traffic engineering and QoS.

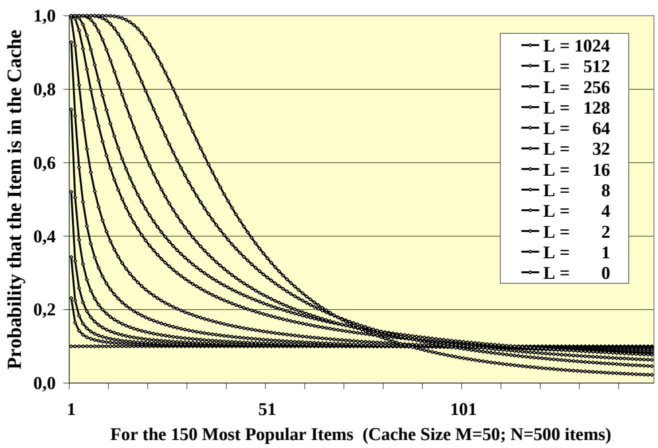

Figure 7: Statistics preferring most popular items in the cache

We studied scalable approximate analysis models for usual caching strategies assuming i.i.d. accesses. Rather than having the exact view on all combinations of items, most of them are aggregated for a unique and tractable analytical treatment. The results show that hit rates for statistics based caching strategies are near optimum especially for small caches under skewed Zipf-like access pattern and already for moderate window size. Larger samples can improve the hit rates for medium to large caches. LRU hit rates are typically 10-20% below the assignment of most popular items with largest relative gaps for small caches. The analysis of hit rates can support the planning of cache locations within a network provider platform.

For future work, more detailed evaluations and alternative strategies can be considered. A usual variant of adaptive access statistics is to introduce a weighting factor that is geometrically decreasing with the backlog. The analysis approach is transferable to this case starting from different distribution functions for the ranking of items. Another topic for future research is the modeling of dynamically changing access popularity, where high dynamics detracts from caching efficiency. Studies on file sharing indicate that the ranking of most popular content is only slowly varying on time scales of weeks or months but more measurement on the access dynamics for content delivery platforms is required.

References

[1] B. Ager, F. Schneider, J. Kim and A. Feldmann, Revisiting Cacheability in Times of User Generated Content, Proceedings of the 13th IEEE Global Internet SymposiumSan Diego, CA, USA (2010)

[2] V. Agrawal, A. Feldmann and C. Schneideler, Can ISPs and P2P users cooperate for improved performance? ACM SIGCOMM Comp. Comm. Review 37 (2007) 31-40

[3] Akamai, State of the Internet, <www.akamai.com> (2010)

[4] Australian Bureau of Statistics, Internet activity (2010) <http://abs.gov.au/ausstats/abs@.nsf/mf/8153.0>

[5] G. Barish and K. Obrazcka, World wide web caching: Trends and techniques, IEEE Communications Magazine (May 2000) 178-185

[6] R. Bolla et al., A measurement study supporting P2P file-sharing community models, Computer Networks 53 (2009) 485-500

[7] L. Breslau et al., Web caching and Zipf-like distributions: Evidence and implications, Proc. IEEE Infocom (1999)

[8] M. Cha et al., I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system, Internet measurement conference IMC'07, San Diego, USA (2007)

[9] J. Charzinski, Traffic properties, client side cachability and CDN usage of popular web sites, Proc. 15th MMB conference, Essen, Germany, Springer LNCS 5987 (2010) 182-194

[10] X. Cheng, C. Dale and J. Liu, Statistics and social network of YouTube videos, IEEE Proc. International Workshop on Quality of Service (IWQoS), Twente, The Netherlands (2008) 249-258

[11] Cisco Systems, Approaching the Zettabyte Era / Global IP traffic forecast and methodology, White paper series (2009) <www.cisco.com>

[12] K. Cho, K. Fukuda, H. Esaki, A. Kato, The impact and implications of the growth in residential user-to-user traffic, ACM Sigcomm Conf., Pisa (2006) <www.acm.org/sigs/sigcomm/sigcomm2006>

[13] A. Dan and D. Towsely, An approximate analysis of the LRU and FIFO buffer replacement schemes, SIGMETRICS Perform. Eval. Rev. 18 (1990)143-152

[14] M. Eubanks, The video tsunami: Internet television, IPTV and the coming wave of video on the Internet, Plenary talk, 71. IETF meeting (2008) <www.ietf.org/proceedings/08mar/slides/plenaryt-3.pdf>

[15] E. Gelenbe, A Unified Approach to the Evaluation of a Class of Replacement Algorithms, IEEE Trans. on Computers 22 (1973) 611-618

[16] P. Gill, M. Arlitt, Z. Li and A. Mahanti, YouTube traffic characterization: A view from the edge, Internet measurement conference IMC'07, San Diego, USA (2007)

[17] L. Guo et al., Does Internet media traffic really follow Zipf-like distributions? ACM SIGMETRICS (2007)

[18] G. Haßlinger, ISP platforms under a heavy peer-to-peer workload, in Peer-to-Peer Systems and Applications, Eds.: R. Steinmetz and K. Wehrle, Springer LNCS 3485 (2005) 369-382

[19] G. Haßlinger, Cross-Layer Aspects of P2P Overlays on ISP infrastructures, IEEE Proceedings P2P'08, 8. International Conf. on P2P Computing, Aachen, Germany (2008) 173-176

[20] G. Haßlinger et al.: Measurement and characteristics of aggregated traffic in broadband access networks, Proc. 20. Internat. Teletraffic Congress, Ottawa, Canada, Springer, LNCS 4516 (2007) 998-1010

[21] G. Haßlinger, F. Hartleb and T. Beckhaus: User access to popular data on the Internet and approaches for IP traffic flow optimization, Proc. ASMTA Conf., Madrid, Spain, Springer LNCS 5513 (2009) 42-55

[22] C. Huang, A. Wang, J. Li, K. Ross, Understanding Hybrid CDN-P2P, Proc. NOSSDAV Conf., Braunschweig, Germany (2008) 75-80

[23] IEEE 802.3 Higher Speed Study Group <ieee802.org/3/hssg> (2009)

[24] Internet Engineering Task Force (IETF), ALTO WG: Application layer traffic optimization, <www.ietf.org/html.charters/alto-charter.html>

[25] W.F. King III, Analysis of paging algorithms, Proc. 5. IFIP Congress, Ljubjana, Yugoslavia (1971)

[26] A. Mahanti, C. Williamson, and D. Eager, Traffic Analysis of a Web Proxy Caching Hierarchy. IEEE Network, 14(3) (2000) 16-23

[27] MINTS: Minnesota Internet traffic studies (2010) <http://www.dtc.umn.edu/mints/references.html>

[28] J. Mogul, Y. Chan and T. Kelly, Design, Implementation and Evaluation of Duplicate Transfer Detection in HTTP, Proceedings 1. Symposium on Network Systems Design and Implementation (2004) 43-56

[29] A. Odlyzko, Internet traffic growth: Sources and implications, Proc. SPIE Vol. 5247 (2003) 1-15

[30] Office of the Telecommunications Authority (OFTA) of the Hong Kong Special Administrative Region, Statistics of Internet Traffic Volume <www.ofta.gov.hk/en/tele-lic/operator-licensees/opr-isp/s2.html> (2010)

[31] O. Saleh and M. Hefeeda, Modeling and caching of P2P traffic, Proc. IEEE Internat. Conf. on Network Protocols (2006)

[32] J. Schroeder-Bernhardi, Analysis of the communication and traffic in P2P networks including web caches, Master Thesis, KOM-D-260, Univ. of Darmstadt, Germany (2006)

[33] D. Starobinski and D. Tse, Probabilistic Methods for Web Caching, Performance Evaluation 46 (2001) 125-137

[34] A.-J. Su, D.R. Choffnes, A. Kuzmanovic and F.E. Bustamante, Drafting behind Akamai, Proc. ACM SIGCOMM, Pisa, Italy (2006)

[35] H. Xie et al., P4P: Provider Portal for Applications, Proc. ACM SIGCOMM, Seattle, USA (2008)