Performance Analysis of Local Caching Replacement Policies for Internet Video Streaming Services

Jie Li[1]
[1]Networking and Transmission Laboratory
Acreo Swedish ICT AB, Kista, Sweden
Email: jie.li@acreo.se

Jinlong Wu[2], György Dán[2]
[2]School of Electrical Engineering
KTH Royal Institute of Technology
Stockholm, Sweden

Åke Arvidsson[3]
[3]Ericsson AB, Stockholm, Sweden

Maria Kihl[4]
[4]Department of Electrical and Information Technology
Lund University, Lund, Sweden

Abstract*—In this work, the performance of 5 representative caching replacement policies was investigated and compared for caching Internet video-on-demand (VoD) in local access networks. Two measured traces of end-user requests were used in the analyses for two typical VoD services: TV-on-demand and user generated content represented by YouTube. The studied policies range from simple *least recently used (LRU)* and *least frequently used (LFU)* algorithms to more advanced ones denoted as *LFU-dynamic lifespan (LFU-DL), Adaptive replacement cache (ARC)* and *Greedy-dual size frequency (GDSF)*. Our results show that the ARC policy always outperforms the other policies due to its adaptive nature and its ability to track changes in the traffic patterns. On the other hand, the simple LRU policy can also achieve a caching performance which is comparable to that of the more advanced ARC policy especially for the TV-on-demand service when the potential caching gain is high. On the contrary, the simple LFU policy always shows the poorest performance. However, by applying a proper lifespan supplement under the LFU-DL policy, the caching performance can be effectively enhanced to the level achievable using ARC and LRU policies. Moreover, the GDSF policy does not outperform simple LRU or LFU-DL, especially for YouTube video clips when the potential caching gain is relatively low. The advantage of GDSF manifested in our analysis is, however, its outstanding cache space usage efficiency among the five studied caching algorithms.

Keywords—Video-on-demand, cache, caching replacement policy, caching algorithm, TV-on-demand, YouTube, local access network

I. Introduction

Internet has now become the global information sharing and communication medium covering every aspect of people's work and life. Even so, global Internet traffic keeps on growing steadily with the ever-increasing, mainly video-oriented content distribution services, especially consumer video-on-demand (VoD) traffic that will contribute to nearly 70% of all consumer Internet traffic in 2017 (excluding video exchanged through peer-to-peer (P2P) file sharing) [1-2]. Compared with conventional web content, rapidly growing video streaming media services require much more bandwidth and can lead to serious network congestion in the backbone network. In addition, the increasing popularity of high definition videos also puts a heavy burden on remote VoD content servers and further increases traffic volumes on the already heavily loaded backbone network. This creates challenges for network operators and Internet Service Providers (ISP) to meet the quality-of-service (QoS) requirements of multimedia services needed to provide sufficient quality-of-experience (QoE) to the end users [3].

One common method to offload backbone network video traffic is to place cache servers locally at the edge of broadband access networks that store popular video contents so that end users can access these videos locally instead of fetching them from remote content servers. Accordingly the burden on the original content servers and the heavy traffic load on backbone networks can effectively be alleviated, and in the meantime the content fetching latency can also be improved [4-9]. Indeed, with the increasing popularity of various Internet VoD services, e.g. the so-called user generated content (UGC) video services such as YouTube or TV-on-demand services, much research has been carried out to study video streaming traffic patterns and user behavior in order to investigate the potential of local network caching gains [10-14]. The results show that online streaming video services usually have significant traffic locality and hence good potential local network caching gains if local network cache servers with an unlimited cache size were employed [11-13]. Nevertheless, as traffic patterns of different VoD services are dependent on the types of video content, potential local network caching gains can vary significantly. For example, for TV-on-demand services, the cacheability of the video content is in general very high (over 90%) [13] as compared to a UGC service such as YouTube [11-12]. In addition, video clips of different categories and durations can also have great impact on the local network caching performance [12-13]. Furthermore, it should be noted that although the advantages of local network caching are obvious, caching segmented video contents in local network servers may cause stream quality oscillations and hence worse QoE to the end users under adaptive video streaming schemes due to the bandwidth mismatch between the access and backhaul networks [14].

In this work, we go beyond conventional analyses of the potential of caching video in local networks and study caching performance using cache servers with limited caching capabilities under different caching replacement policies. Previously, extensive investigations had been carried out to study the performance of caching replacement policies for conventional Web browsing [4]. In the meantime, research on caching VoD content has either been focused on the optimization of local network VoD cache architectures [5-6], or segment-based caching schemes in order to address the large content size of VoD services [7], while usually a simple (e.g. *least recently used (LRU)*) caching algorithm was used to estimate the corresponding performance of the proposed solutions. Studies on particular caching replacement policies were also carried out using simulated VoD request patterns due to the lack of measured data traces in real environments [8-9]. Different from these previously reported studies, in this paper, we report a systemic study on the local caching performance of five representative caching replacement policies based on measured end-user video request traces in two local access networks in Sweden for two typical VoD services: TV−on−demand and YouTube. The caching replacement policies studied range from simple *LRU* and *least frequently used (LFU)* algorithms to the more advanced *LFU-dynamic lifespan (LFU-DL), adaptive replacement cache (ARC)* and *greedy-dual size frequency (GDSF) policies.* To the best of our knowledge, this is the first systematic evaluation and comparison of the performance of different caching replacement policies for VoD services using measured end-user request data traces.

Our major findings are:

- The ARC policy always outperforms the other policies due to its capability to track changes in the traffic pattern over time.
- For the simple and common LRU and LFU policies the caching performance is rather different, i.e. LRU can achieve a performance comparable to the more advanced ARC policy with high potential local network caching gains, especially for the TV-on-demand service, while the LFU policy always shows the poorest performance.
- By applying a proper lifespan supplement to LFU, called the LFU-DL policy, the caching performance can be effectively enhanced to the level achievable by ARC and LRU. The results verify the intuitive understanding of Internet consumer VoD traffic that in general newly released videos tend to be more popular than old videos.
- For the GDSF policy, the cache hit rate does not outperform the simple yet efficient LRU or LFU-DL policies, especially for YouTube video clips for which the potential local network caching gain is relatively low. The advantage of GDSF is, however, its outstanding cache space efficiency among the 5 studied caching algorithms.
- Analysis on the performance of the caching algorithms under a video category based separate caching scheme for the TV-on-demand service indicates that caching performance differs significantly with the video categories, while in general the relative performance difference between the cache replacement polices largely remains the same for all the video categories. In the meantime, the overall local network caching performance of the separate caching scheme does not outperform that of a category indiscriminate single cache if the separate category cache sizes are set based on the category unique videos per day. Thus, further studies on how to assign the local cache space among the different video categories is needed in order to enhance the overall local network caching performance under the video category based separate caching scheme.

The rest of this paper is organized as follows. Section II describes the two VoD data traces used in the analysis, together with an overview of the five representative cache replacement policies. In addition, the definitions of the caching performance parameter (*cache hit rate*) and single cache versus separate cache models are also given. In Section III detailed analysis results are presented and discussed. Finally in Section IV the conclusions are drawn.

II. METHODOLOGY

A. *Data traces*

Two measured end-user video watching request data traces in two local access networks in Sweden were used in this work: one is a trace of requests to a commercial TV−on−demand service, while the other one is a trace containing requests for YouTube videos.

1) *Data trace of TV-on-demand service*

The TV-on-demand data trace was collected from a commercial over-the-top (OTT) Internet TV-on-demand service in an urban area in Sweden over 11 weeks from 2012−12-31 to 2013-03-18 covering a total of 13437 unique viewers. In the data trace, each request record contains the parameters of *video_title, (anonymous) viewer_id, start_time,* and *viewing_time*. Besides, the TV video clips are categorized under 7 major categories, namely *Children*, *Documentary*, *Home & Hobby*, *News & Debate*, *Entertainment*, *Sport* and *TV Series*. The video quality (resolution and/or video size) parameters are not available in the data trace, hence the maximum viewing time of each video is regarded as a reference of video size with the assumption that all videos had the same resolution. Table I summarizes the general statistics of the total data set including the indicative parameter of the theoretical potential total local network caching gain defined as $(\textit{total requests} - \textit{unique videos})/\textit{total requests}$ (assuming a cache server with unlimited cache size). Table II summarizes the viewing statistics under each category, in which the relative contribution to total potential local network caching gain is basically the share of repeated requests (i.e. *total requests – unique videos*) of a category among the total repeated requests [12]. From Table II one can see that videos of *Entertainment*, *Sports*, *TV-Series* and *Home & Hobby* dominated the content of the TV-on-demand service, while videos of *Children*, *News & Debate* and *Documentary* played less significant roles.

2) Data trace of YouTube

The YouTube UGC data trace was extracted from a 3 week YouTube request traffic packet dump between 2012–06–08 and 2012–07–05 in another Swedish municipal residential broadband access network with approximately 2600 households connected to the network. The details of the data trace extraction were described in [12]. Table III gives the overview of the extracted YouTube data trace statistics over the 3 week period of time, including the indicative parameter of the theoretical potential total local network caching gain.

TABLE I. STATISTICS OF TV-ON-DEMAND DATA

Total requests	Unique videos	Unique viewers	Potential total local network caching gain*
244816	13936	13437	94.3%

*Potential total local network caching gain: *(total requests - unique videos)/total requests*

TABLE II. STATISTICS OF TV-ON-DEMAND DATA BY VIDEO CATEGORY

Category	Total requests	Unique videos	Potential local caching gain	Relative contribution to total potential local network caching gain
Children	7531	873	88.4%	2.9%
Documentary	30449	2218	92.7%	12.2%
Home & Hobby	46310	1607	96.5%	19.4%
News & Debate	9132	3273	64.2%	2.5%
Entertainment	53663	2958	94.5%	22%
Sport	41780	1975	95.3%	17.2%
TV Series	55803	975	98.3%	23.8%

TABLE III. CAPTURED YOUTUBE REQUEST DATA TRACE OVERVIEW

Total requests	Unique videos	Unique viewers	Potential total local network caching gain
133941	89380	6321	33.3%

B. Cache replacement policies

For realistic local network cache servers with limited caching capabilities, suitable replacement policies are required to decide whether a piece of content should be inserted into the cache or not and, if the cache is full, which content should be removed to make space. In general, the most important factors that affect the caching efficiency (usually denoted as *cache hit rate* as described in the following) include recency, frequency, size, fetching cost and modification/expiration time of cached content [4]. Accordingly, cache replacement policies can be classified into recency, frequency or their combination based policies, as well as other 'functionized' policies taking into account factors like content size, fetching cost and expiration time etc.

In this work, five representative cache replacement policies covering the most important factors that influence caching performance are studied to investigate the local network caching performance for online OTT VoD services:

- **Least recently used (LRU)**, the most widely used and easily implementable cache replacement policy that replaces the least recently used content.

- **Least frequently used (LFU)**, also widely used to replace the least frequently used content.

- **LFU-dynamic lifespan (LFU-DL)**. The LFU policy may suffer a so-called "cache pollution" problem, i.e., videos that accumulate a large number of requests (usually during a short period of time) may not be accessed again, but will still occupy the cache space under the LFU policy, forcing videos with more recent uses to be evicted instead. As a result, cache hit rate may be severely decreased [15]. Dynamic aging is a simple and effective complement for the LFU policy to deal with cache pollution. With dynamic aging, every cached video is assigned a lifespan which will be refreshed upon every hit of the cached video. When the lifespan of a cached video expires (with no new refreshment), the video will be evicted no matter whether the cache is full or not.

In order to determine a suitable lifespan for cached videos, the replay interval distribution of the video content can be used to identify a turning point in the distribution curve that covers most, e.g. 80% of the replays, and this turning point can be used as a reasonable estimation for the video lifespan.

- **Adaptive replacement cache (ARC)**. The ARC policy uses the history of recently evicted content to change its recency or frequency preferences. It is thus a combination of LRU and LFU. In more detail, the ARC policy splits the cache into two parts, *T1* and *T2*. *T1* caches the contents that only have been accessed once, and *T2* caches the contents that have been accessed many times. Besides, this policy maintains two other lists, *B1* and *B2* to record the (LRU-based) eviction history of *T1* and *T2*, respectively. Recency or frequency preferences are adjusted by dynamically changing the target sizes of *T1* and *T2* according to the eviction histories recorded in *B1* and *B2*. In this way, the ARC policy can trace changes in traffic patterns and adjust the replacement policy to emphasize frequency or recency accordingly [16].

- **Greedy-dual size frequency (GDSF)**. Apart from recency and frequency, this cache replacement policy also takes video clip size and fetching cost into consideration by assigning a 'priority key', P_r, to each content file defined as

$$P_r = clock + f \times \frac{C}{S} \qquad (1)$$

where the parameter *clock* is a monotonically increasing queue 'clock' starting from 0 and will be updated when a replacement occurs with the priority key of the replaced content file. In addition, *f* is the access frequency or request count, *C* is the fetching cost, and *S* is the size of the content file, respectively. Under this policy, if a content file in a full cache has a lower priority key than a newly arrived one, cache replacement occurs. Obviously this cache replacement policy favors small size content files in order to increase the cache hit rate (in terms of objects), in parallel to the access frequency and the recency (aging) of the cached content (through the monotonically increasing parameter *clock*) [17-18].

In the analyses of this work using GDSF, due to the lack of relevant parameters, the fetching cost was set to unit for all videos. In addition, the file sizes were estimated based on the durations of the clips, assuming that all videos in the studied data traces have the same resolution.

C. *Cache hit rate*

The performance of different cache replacement policies is measured by the *cache hit rate*, defined as $H = T/R$, where T denotes number of successful hits in the local network cache server and R denotes the total number of requests. Since the cache hit rate is computed based on the number of requests, it corresponds to the *object hit rate*.

D. *Single cache model versus separate cache model*

In addition to analyzing the caching performance using a single network cache, it might also be interesting to further investigate the caching performance under a separate per–category cache scheme. In the separate cache model, videos of the same category are cached independently, i.e., a newly arrived video will only replace the video in the same category, while for the single cache model, all the videos are stored in the same cache and the replacement of videos in the cache does not consider the video category.

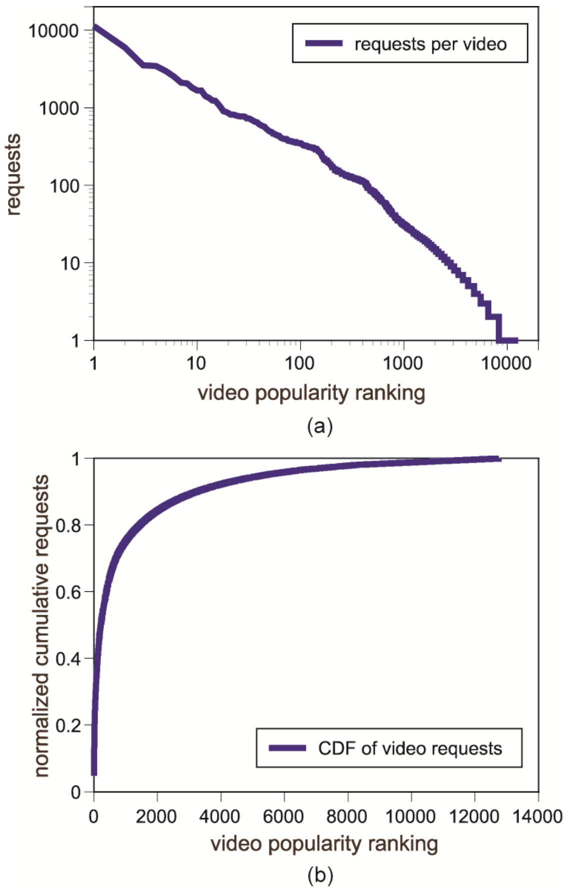
III. RESULTS AND DISCUSSIONS

The primary focus in this work is analyzing the performance of caching replacement policies for the TV–on–demand service using the measured eleven weeks data trace, while analysis for the corresponding three weeks YouTube data trace was carried out as a comparison. The cache replacement policies were implemented with Python scripts.

A. *Caching performance of TV-on-demand service*

1) Video popularity and cumulated video request distributions

Fig. 1 shows the video popularity in terms of requests vs. popularity rank (in log-log scale) and the corresponding cumulative distribution of requests per TV video when ordered by popularity rank. We can see that for the TV–on–demand service, the video watching popularity is highly skewed, with 13.8% of all the requested TV videos accounting for 80% of all the requests. This observation is in accordance with the so-called Pareto rule, also denoted as "80–20" rule that about 20% content will account for roughly 80% total requests [13]. This skewedness of video popularity forms the basis for caching using limited-sized caches irrespective of whether the distribution of requests actually follows a Zipf-like distribution [13].



(a)

(b)

Fig. 1. Video popularity (a) and normalized cumulative distributions of video watching requests (b) for TV-on-demand service.

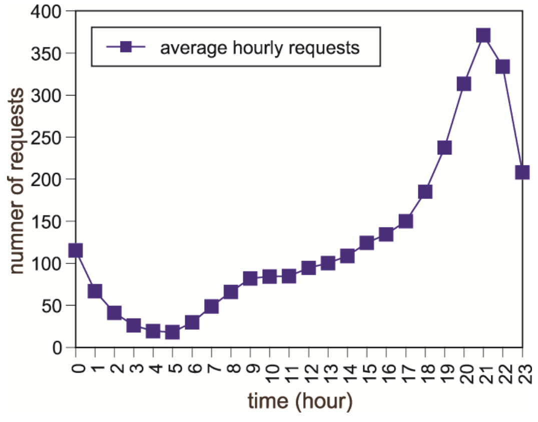


Fig. 2. Average requests per hour of TV-on-demand service.

Fig. 2 shows the hourly distribution of the number of requests of the TV-on-demand service, and exhibits the typical periodic 24-hour time distribution with the "prime time" between 19:00 and 22:00 in the evening. This periodic time distribution also suggests that a single or multiple 24–hour-based lifespan might be a simple yet reasonable estimation to the cached videos in the LFU-DL caching algorithm. Indeed, a further analysis of the replay interval distribution revealed that 78.5% of all the replays occurred within one day. Therefore, based on this observation, a one–day lifespan was applied in the implementation of the LFU-DL caching algorithm. This has proved to be a reasonable choice to achieve comparable caching performance as other more advanced caching algorithms e.g. ARC (see results in the next subsection).

2) *Performance of cache replacement policies*

The sizes of our limited caches were chosen as certain percentages of the maximum number of unique videos per day (for each category as well as for the whole data set), as illustrated in Table IV that shows the sizes (in terms of number of videos) of the caches when set to 10% and 20% of the maximum number of unique videos per day respectively.

TABLE IV. DAILY UNIQUE VIDEOS

Category	Maximum daily unique videos	10% of daily unique videos	20% of daily unique videos
Children	70	7	14
Documentary	230	23	46
Home & Hobby	230	23	46
News & Debate	70	7	14
Entertainment	200	20	40
Sport	80	8	16
TV Series	120	12	24
All categories	1000	100	200

a) Single cache performance

We first study the caching performance using a single network cache for all video categories. Fig. 3 shows the obtained cache hit rates at different cache sizes for the 5 cache replacement policies. We see that, first of all, the ARC policy always outperforms the other policies thereby demonstrating the superb capability of its adaptive nature. Moreover, interestingly enough, the simple LRU policy also shows good performance which, in fact, is slightly better than the performance of the more advanced GDSF policy. On the contrary, the equally simple LFU caching algorithm shows poor cache hit rates even when the cache size was increased to 25% of total daily unique videos. This is attributed to the cache pollution problem as mentioned earlier. However, by simply assigning a 24-hour lifespan to the videos stored in the local cache under the LFU-DL policy, a performance comparable to the more advanced algorithms can be achieved at cache sizes over 10% of daily unique videos. This also verifies that a lifespan of 24-hours indeed is a reasonable estimation for TV-on-demand videos.

Another observation in Fig. 3 is that GDSF does not outperform simpler algorithms such as LRU or LFU-DL, and its cache hit rate drops even faster than that of ARC and that of LRU at smaller cache sizes (less than 10% of total daily unique videos). However, compared to other caching replacement policies, GDSF has the advantage that it tends to cache small video files hence, for a given cache size measured in bytes or the total length of stored videos, GDSF will store more videos, or equivalently if the cached number of videos is fixed, GDSF will use the least cache space. As an illustration, Fig. 4 (a) shows the maximum cache sizes measured in total minutes of cached video (obtained during the caching simulation process) for different caching algorithms when the cache size was set at 10% of the number of unique videos per day (i.e. to 100 videos). We see that GDSF indeed features a

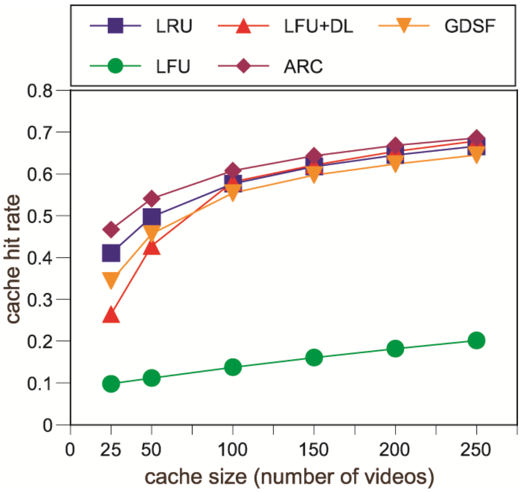


Fig. 3. Cache hit rate versus cache sizes for the different caching replacement policies using a single local network cache.

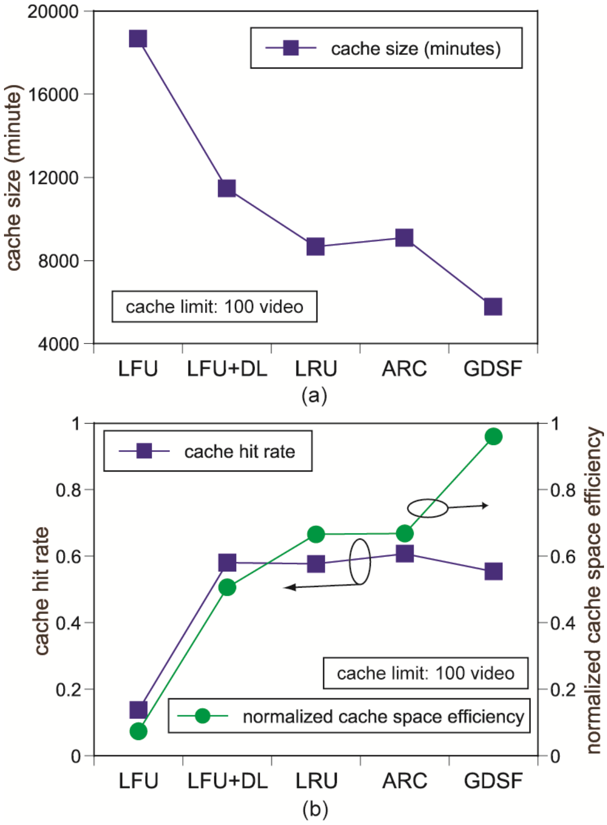


Fig. 4. Maximum cache size (in total minutes of cached video) (a) and normalized cache space efficiency (b) for different caching algorithms at 100 video cache limit, respectively.

smallest cache size measured in the total cached video length (i.e., 5772 minutes, or ~ 216 Gigabytes if we assume a video quality with the bit rate of 5 Mbit/s), as compared to e.g. LFU with a maximum cached video length up to 18662 minutes (or ~ 700 Gigabytes with the same video quality). Moreover, if we further link the size of the cache to the corresponding cache hit rate, we can define a parameter of *normalized cache space efficiency* (per 10000 minutes cache space) as

$$\frac{cache\ hit\ rate}{cache\ size\ (minute)} \times 10000 \qquad (2)$$

Fig. 4(b) shows the calculated normalized cache space efficiency for different caching algorithms at the same cache size as in Fig. 4(a). For comparison, the corresponding cache hit rates are also shown. We can see clearly that GDSF has a significantly better cache space efficiency compared to other caching replacement policies. It is therefore preferable to use GDSF if cache space is a limiting factor or if smaller caches are favored in order to achieve e.g. higher energy efficiency in local network caching schemes.

b) Separate cache performance per video category

From Table II we see that for the Video-on-demand service, videos classified as *Entertainment*, *Sports*, *TV-Series* and *Home & Hobby* dominated the contribution to the overall potential local network caching gain, while videos classified as *Children*, *News & Debate* and *Documentary* played less significant roles. It may thus be interesting to see how the performance of the caching algorithms varies for different video categories and how this influences the overall caching performance. To do so, we adopted a *separate cache model*, i.e., video replacement in the cache only occurred within the same video category. Besides, to be consistent with the criterion of setting the cache sizes using a single network cache the maximum number of cached videos in a category was set according to its share of the maximum number of unique videos per day as shown in the third column of Table IV. Fig. 5 shows the results obtained by setting the cache size to 10% of the maximum number of unique videos per day. From the figure we see that, for the same cache size limit, videos of *Entertainment*, *Sports*, *TV-Series* and *Home & Hobby* have significantly higher cache hit rates than those of *Children*, *News & Debate* and *Documentary*, suggesting that the viewers' interests in the dominating video categories (*Entertainment*, *Sports*, *TV-Series* and *Home & Hobby*) are more concentrated and hence easier to cache. Moreover, the hit rate variations for each category among different caching replacement policies are similar to the results under the single cache model shown in Fig. 3, with the exception for the *sport* category where the LFU policy performs relatively better than other categories. The reason for this is that newly released popular *sport* videos can accumulate a large amount of repeated requests over a very short period of time, resulting in high request frequency for most recent video clips. The other observation in Fig. 5 is that for the *TV Series* category, the dynamic lifespan supplement of the LFU policy does not improve the cache hit rate as significantly as for other categories, which is attributed to a slower request decay rate than for other categories.

c) Separate cache versus single cache

With the separate cache scheme described above, one may wonder about the performance of a per-category caching scheme compared to the single cache scheme. Fig. 6 shows the result of the comparison between the two caching schemes at the cache sizes of 10% and 20% of the maximum number of unique videos per day (corresponding to the third and fourth column in Table IV), respectively. We see that on the one hand, the overall caching performance of the two caching schemes is quite close for LRU, ARC and GDSF caching algorithms especially for the larger cache size (200 videos), even though the single cache scheme has slightly better performance (which is attributed to the popularity difference among the video categories as discussed in the following). On the other hand, for simple LFU, the separate cache scheme has significantly better performance than the single cache scheme. Nevertheless, by adding a dynamic lifespan to the cached videos under the LFU-DL policy, the single cache scheme outperforms the separate cache scheme again.

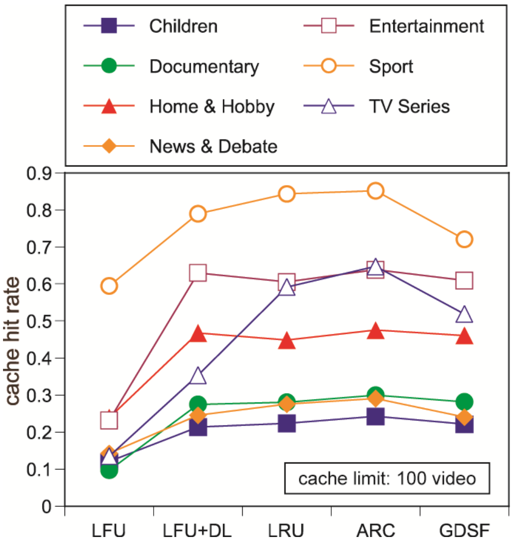


Fig. 5. Cache hit rate of different video categories versus caching replacement policies under separate cache model at 100 (total) video cache limit.

As mentioned earlier, in this work, the separate cache sizes were chosen based on the unique video numbers of each category per day, in accordance with the criterion of setting the cache sizes using a single local network cache. However, if we compare the share of daily unique videos of a category to the corresponding relative contribution to the potential local network caching gain of the same category, as shown in Table V, we can see immediately that for the dominating categories, especially *Sports* and *TV-Series*, the relative contributions to the potential total caching gains are two times higher than their corresponding unique video shares, while for *Documentary* the relationship is inversed. This observation suggests that using the daily unique video shares to set the separate cache sizes is not an optimal choice in respect of enhancing the overall network caching gain, as it disregards the relative contributions of the video categories to the total network caching gain. A more sophisticated scheme could be formulated as a constrained optimization problem, and would set the separate cache sizes based on the categories' marginal contributions to the caching gain. Such a policy will be subject of our future work.

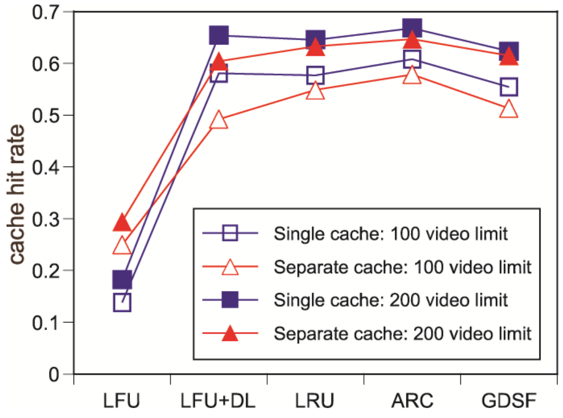


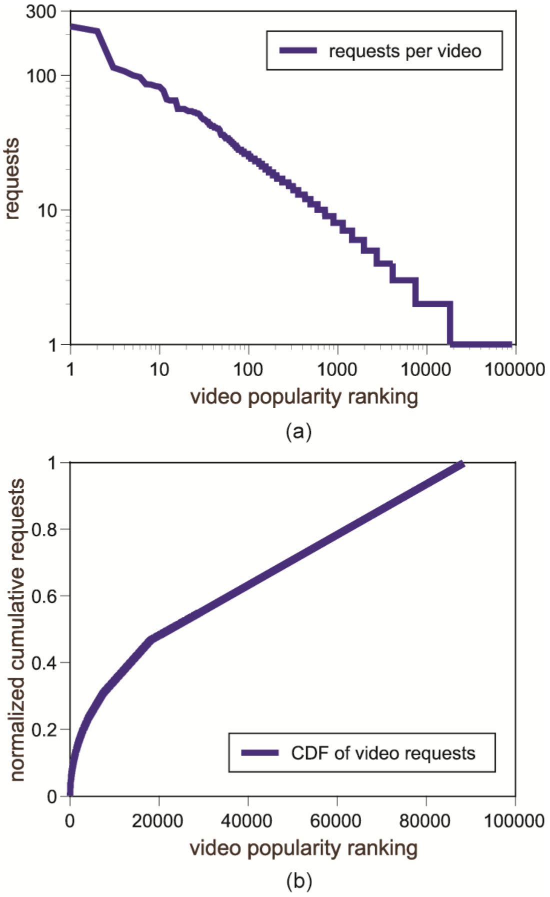
Fig. 6. Single cache versus separate cache schemes.

TABLE V. CATEGORY DAILY UNIQUE VIDEO SHARE VERSUS RELATIVE CONTRIBUTION TO POTENTIANL LOCAL NETOWRK CACHING GAIN

Category	**Daily unique video share**	**Relative contribution to total potential local network caching gain**
Children	7%	2.9%
Documentary	23%	12.2%
Home & Hobby	23%	19.4%
News & Debate	7%	2.5%
Entertainment	20%	22%
Sport	8%	17.2%
TV Series	12%	23.8%
All categories	100%	100%

B. Caching performance of YouTube

From Table I and Table III one can see that YouTube content has significantly lower potential caching gain compared to TV-on-demand due to the large number of unique YouTube video clips. Fig. 7 shows the video popularity in terms of requests vs. popularity rank (in log-log scale) and the corresponding cumulative distribution of requests per the YouTube video when ordered by popularity rank. We can see that, compared to TV-on-demand shown in Fig. 1, the skewedness of video watching popularity is significantly reduced, with 20% of the most popular YouTube video clips accounting for only about 40% of all the requests. In addition, analysis of the replay interval distribution of the YouTube data trace showed that a 3-day period of time covers slightly over 80% of all replays, indicating that a 3-day lifespan is a more suitable choice for the YouTube video clips (as compared to the 1-day lifespan for the TV-on-demand service).



(a)

(b)

Fig. 7. Video popularity (a) and normalized cumulative distributions of video watching requests (b) for UGC-YouTube service.

With all these observations in mind, it is thus interesting to investigate the corresponding local network caching performance for YouTube using realistic cache servers with limited sizes and compare the results with those of TV–on–demand. Fig. 8 shows the results for the single cache scheme for the YouTube data trace and different cache sizes (in terms of number of videos). To compare the results to those in Fig. 3 note that the number of unique videos per day, which amounts to 1000 for the TV-on-demand trace, amounts to about 4000 for the YouTube trace. We see that, the two traces are similar in that the ARC policy outperforms the other caching schemes and in that the simple LFU policy has the poorest performance, while a dynamic lifespan supplement under the LFU-DL policy again can improve the performance dramatically. However, the relative performance differences between the studied caching algorithms become significantly larger, especially the difference between the simple LRU policy and the more advanced GDSF policy, which shows significantly poorer hit rates (relative to ARC and LRU) than for the TV-on-demand trace. This is attributed to the fact that YouTube video clips mostly are short videos with 90% of the video clips in the data trace less than 15 minutes [12]. Consequently, the advantage of GDSF (by increasing the cache hit rate through caching more small files) becomes less significant, making it less preferable for caching short duration clip dominated YouTube videos.

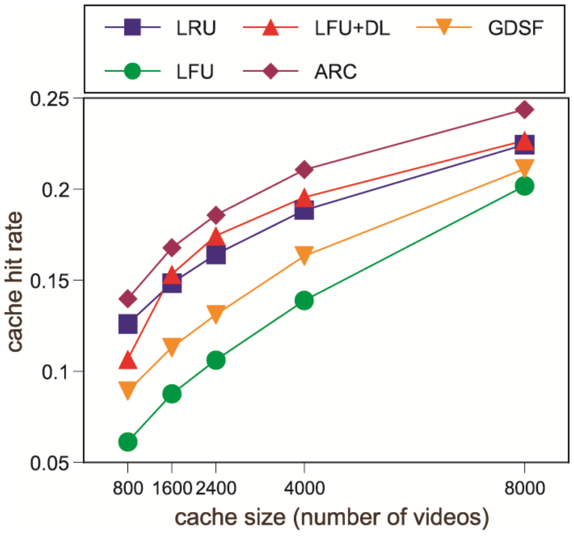


Fig. 8. Cache hit rate versus cache sizes for the different caching replacement policies for YouTube video service.

IV. CONCLUSIONS

In this work, the performance of five representative cache replacement policies was investigated for caches of limited size in the access network for Internet VoD services. Two measured traces of end-user video requests were used in the analyses for two typical VoD services: TV-on-demand and YouTube as an example of user generated content. Our results show that the ARC policy always outperforms the other policies due to its capability to track changes in the traffic pattern over time. Nevertheless, the drawback of the ARC policy is the implementation complexity and longer processing time, which can be decisive in selecting replacement policies for content caching. For the simple and common LRU and LFU policies the caching performance is rather different, i.e. LRU can achieve a performance comparable to the more advanced ARC policy with high potential local network caching gains, especially for the TV-on-demand service, while the LFU policy always shows the poorest performance. However, by applying a proper lifespan supplement to LFU, called the LFU-DL policy, the caching performance can be effectively enhanced to the level achievable by ARC and LRU. These results verify the intuitive understanding of Internet consumer VoD traffic that in general newly released videos tend to be more popular than old videos. Moreover, for the GDSF policy, the cache hit rate does not outperform the simple yet efficient LRU or LFU-DL policies, especially for YouTube video clips for which the potential local network caching gain is relatively low. The advantage of GDSF is, however, its outstanding cache space efficiency among the 5 studied caching algorithms.

ACKNOWLEDGMENT

This work was supported by the Swedish Governmental Agency for Innovation Systems (Vinnova) in the European CelticPlus project NOTTS and in the project EFRAIM, and through the Center for Networked Systems (CNS) under the Institute Excellence Center Program.

REFERENCES

[1] J. Li, A. Aurelius, V. Nordell, M. Du, Å. Arvidsson, and M. Kihl, “A five year perspective of traffic pattern evolution in a residential broadband access network”, Future Network & Mobile Summit 2012, July 2012, Berlin, Germany

[2] White Paper, “Cisco Visual Networking Index: Forecast and Methodology, 2012–2017”, May 2013, Cisco

[3] IP Network Monitoring for Quality of Service Intelligent Support (IPNQSIS), http://projects.celtic-initiative.org/ipnqsis/

[4] S. Podlipnig, and L. Böszörmenyi, "A survey of web cache replacement strategies", ACM Computing Surveys, Vol. 35, Issue 4, Pages 374-398, December 2003

[5] S. Acharya and B. Smith, “*MiddleMan: a video caching proxy server*”, Proceedings of NOSSDAV, June 2000

[6] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, “Impact of traffic mix on caching performance in a content-centric network”, 2012 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pages 310 – 315, March 2012

[7] K. Wu, P. S. Yu and J. L. Wolf, “Segment-based proxy caching of multimedia streams”, Proceedings of the 10th international conference on World Wide Web, pages 36-44 ACM New York, USA ©2001

[8] D. De Vleeschauwer and K. Laevens, "Performance of Caching Algorithms for IPTV On-Demand Services", IEEE TRANSACTIONS ON BROADCASTING, VOL. 55, NO. 2, JUNE 2009

[9] T.R.Gopalakrishnan Nair and P. Jayareka, "A Rank Based Replacement Policy for Multimedia Server Cache Using Zipf-Like Law", JOURNAL OF COMPUTING, VOLUME 2, ISSUE 3, MARCH 2010

[10] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat, "Modeling and Generating Realistic Streaming Media Server Workloads", Computer Networks, Vol. 51, Issue 1, pages 336–356, 2007

[11] Å. Arvidsson, M. Du, A. Aurelius, M. Kihl, "Analysis of User Demand Patterns and Locality for Youtube traffic", 25th International Teletraffic Congress (ITC), Sept. 2013

[12] J. Li, H. Wang, A. Aurelius, M. Du, Å. Arvidsson, and M. Kihl, “YouTube Traffic Content Analysis in the Perspective of Clip Category and Duration”, Fourth International Conference on Network of the Future (NoF'13), Pohang, South Korea, October 2013

[13] H. Abrahamsson and M. Nordmark, "Program Popularity and Viewer Behaviour in a Large TV-on-Demand System", Internet Measurement Conference(IMC' 12), Boston, USA, November 2012

[14] C. Mueller, S. Lederer and C. Timmerer, "A proxy effect analysis and fair adaptation algorithm for multiple competing dynamic adaptive streaming over HTTP clients", Proceedings of the Conference on Visual Communications and Image Processing (VCIP) 2012, San Diego, USA, November 27-30, 2012

[15] R. Ayani, Y. M. Teo and Y. Seen Ng, "Cache pollution in Web proxy servers", Parallel and Distributed Processing Symposium, 2003.

[16] N. Megiddo and D. S. Modha, "ARC: a self-tuning, low overhead replacement cache", USENIX File & Storage Technologies Conference (FAST), March 31, 2003, San Francisco, USA

[17] P. Cao and S. Irani, "Cost-Aware WWW Proxy Caching Algorithms", Proceedings of the USENIX Symposium on Internet Technologies and Systems, Monterey, California, December 1997

[18] L. Cherkasova and G. Ciardo, "Role of Aging, Frequency, and Size in Web Cache Replacement Policies", HPCN Europe 2001, LNCS 2110, pp.114-123, 2001