Experimental Analysis of Caching Efficiency for YouTube Traffic in an ISP Network

Fabrice Guillemin, Bruno Kauffmann, Stephanie Moteau, Alain Simonian
Orange Labs, France,
Email: firstname.lastname@orange.com

***Abstract*—In this paper, we report YouTube traffic measurements from Orange IP backbone network connecting residential customers. We exhibit its salient features in relation to the performance of caching. By examining the file popularity distribution, we show that video requests are highly volatile in that a huge number of files are viewed only a few times; these files are therefore not relevant for caching. Nevertheless, there is a subset of files which are massively viewed by end users and are worth caching. On the basis of this experimental observation, we develop a mathematical model for estimating the efficiency of file caching in the presence of noise traffic composed of those files which are rarely requested and thus "pollute" the cache. We then proceed to trace-driven simulations in order to check the qualitative conclusions derived from the theoretical model.**

I. Introduction

As reported in many recent studies, it is now well-known that the volume of streaming traffic carried over telecommunication networks has exploded in the past few years [1], [2]. Within busy hours, streaming can contribute up to 60 % of global traffic. Excluding IPTV which is mostly delivered via dedicated infrastructures, streaming traffic is then basically composed of video downloads from Video on Demand (VoD) service platforms and from user-produced media platforms such as YouTube; at the moment, WebTV or other TV services (e.g. that delivered by peer-to-peer networks) are still marginal. Understanding the performance of caches dedicated to such traffic is consequently critical for estimating the global performance of caching.

The characteristics of VoD downloads in Orange networks, in particular, has been studied in [3]; we here study traffic generated by User Generated Content (UGC) services. The basic difference between VoD content and UGC available from YouTube platforms is that the catalog size for the former is finite, while the number of videos in various formats for UGC is almost unbounded. This implies that the volatility of content requests is much higher for UGC than for VoD.

The problem of volatility is central for caching. In fact, if a network operator aims at load reduction on critical transmission links (typically, peering links) via caching, it is crucial to determine whether some content items are requested sufficiently often so that it is worth caching them. In addition, such items have to be frequently requested in order not to be pushed out of the cache by other items which are rarely downloaded, but are numerous enough to overflow the cache. These two issues require an in-depth analysis of the dynamics of UGC file requests. We here assume transparent caching, as can be deployed by a network operator, and do not take into account the possible interactions, relays and forwarding that can take place in a Content Delivery Network (CDN). In order to keep our presentation simple, we also restrict our results to the most popular replacement policy, namely the Least Recently Used (LRU) policy, which is known to provide a good performance/complexity tradeoff.

The characteristics of YouTube traffic has been studied in the technical literature in various contexts. In [4], YouTube traffic within a campus network is studied; a detailed analysis is performed where the popularity of files can be well approximated by means of a Zipf distribution with an exponent $\alpha < 1$; it is also noted that the mean size of video file is about 10 MB. Another analysis of YouTube traffic from a campus network can be found in [5], where popularity, file sizes, and request arrival process are analyzed in depth; the authors additionally perform a proxy-caching analysis with various cache capacities and observe hit ratios at most 35 %. Popularity of YouTube files is analyzed in [6] by crawling YouTube websites and collecting meta information about video files. The pattern of requests is, in particular, studied in [7]. Additional precise YouTube traffic characterization can be found in [8]. Traffic matrices of YouTube traffic in an ISP network is analyzed in [9]. Inter-domain implications of streaming traffic are addressed in [10], but this is out of the scope of this paper.

In this paper, as in the above mentioned studies, we investigate YouTube traffic characteristics but we consider a totally different data set. Specifically, we report measurements of YouTube traffic from Orange IP backbone network in France which connects residential customers. The customer population size is thus much bigger than that considered in [4], [5], as well as the number of downloaded files. In addition, the usage of YouTube is somewhat different in that a campus usually clusters a large proportion of students with intensive social interactions among them, whereas residential customers are more diverse in their interests and characteristics. Another possible difference is that the bit rates available to end users may be much lower than those affordable in campus networks, implying customer impatience, quality degradation, file download interruptions, etc. It follows that the volatility of video requests may be much higher in the case of residential customers than that in a campus network.

Beyond the characterization of YouTube traffic in a commercial network, the goal of this paper is to investigate the efficiency of caching for such a traffic pattern. In this respect, the volatility of video requests is a critical issue in that the capacity of the cache has to be large in order to achieve reasonably high hit ratio, and that rarely requested files may have an adverse impact on the caching performance by pushing out of the cache those files viewed a large number of times.

The organization of this paper is as follows: In Section II, we describe the methodology used to measure YouTube traffic in the Orange IP backbone network and we provide file statistics, exhibiting the requests volatility. In Section III, we develop a mathematical model of a cache with noisy traffic. In Section IV, we present trace driven simulations of caches. Some concluding remarks are presented in Section V.

II. YouTube traffic measurements

A. Data set description

We consider statistics of YouTube traffic in the Orange IP backbone network. This network connects residential customers as well as small-medium size company premises to the Internet. Various access technologies can be used at the access (namely FTTH, ADSL, xDSL), and in the backhaul (ATM, GigaBit Ethernet). To perform measurements, we use passive probes located on GigaBit Ethernet links connecting the various DSLAM to the BAS (Broadband Access Server). These probes are sufficiently high in the network so as to observe a large number of customers (up to $50\,000$ per probe). Equipped with packet processing capabilities, the probes can monitor traffic flows of individual customers.

The probes were configured to detect YouTube traffic and to generate a single record line per viewed video. This record includes an anonymized customer identifier, the server address, timestamps of the starting and terminating times of the transmission of a video flow and the associated volume in packets and bytes. The video is identified by a 64 bits ID present in YouTube HTTP request which seems to be a constant and unique identifier for a given video file.

One major difficulty when observing YouTube traffic in an ISP network is that a video file can be transmitted over several TCP connections, even in the case of classical progressive HTTP download [8]. This segmentation of files into pieces (or chunks) prevents us from identifying to which part of a file a TCP connection corresponds to[1]. In fact, such an identification requires either access to the packet payload or, in case of HTTP flows, to infer it from the URL. As either solution was not accessible to us, we could not determine which parts of a video had been requested by a given client, and therefore could not study the impact of file download interruption. We therefore assume in the following that each request requires the whole video, ignoring the fact that the same video can be completely viewed by some customers while only partially by some others.

In order to remedy the problem of file segmentation, some preliminary treatment has been performed on the traffic traces. Specifically, we have aggregated those pieces of dowloaded content with the same YouTube ID and carried by several TCP connections between the same IP addresses and with starting times within an interval of 30 seconds. This procedure allows us to reconstruct information on individual video files. By taking the maximum observed volume for a given file ID, we obtain an upper bound for the volume in bytes of a video file. This may introduce some bias in the video size estimation because this maximum download size can be larger than the actual video size (in case of packet losses, web page reload, etc.). The size of video files observed in the Orange commercial network then appears much greater than that observed in 2007 in [4], which is probably due to both our preliminary treatment assumptions and the evolution of YouTube content in the past few years (full movies, increased resolutions, etc.).

In the following, we show video file statistics of YouTube traffic observed in three towns in France: Bordeaux, Lyon and Paris. The two former are equivalent in terms of connected customers; in Paris, data are aggregated from two probes. Finally, we have also computed statistics by using data collected by all probes installed in the IP backbone network (12 probes in total, covering 25 % of the whole customer footprint). Traffic measurements were performed from April 1^{st} to April 15^{th} 2012.

B. File statistics

In a first step, we report statistics for files viewed by end users as observed by the probes located in Bordeaux, Lyon and Paris. To discriminate files, we introduce a threshold (arbitrarily set to 3) for the number of times that a file is viewed in order to decide whether a file is relevant for caching or not. As shown in the following, the chosen threshold first allows us to classify files and to exhibit some salient features of the file population downloaded by end users, notably the huge number of files which are seen only once or twice and which can consequently be considered as noise with regard to caching.

In Table I, we report file statistics in Bordeaux (the results in Lyon and Paris are similar). Within one day (the 7th day of the measurement period in Table I), the number of files downloaded more than twice is only 24 % of the total number of downloads. When the observation window is enlarged, the proportion of files viewed more than twice increases and reaches 55 %. This simple observation shows that the occurrence of files that we consider worth caching may be dispersed in time. It is also worth noting that the total volume of those files downloaded more than twice rapidly increases (more than 4.7 TB in two weeks). This indicates that the potential gain of caching is significant, but that storage capacities in the cache need also to be substantial.

For the sake of completeness, Table II reports file statistics by using data collected by all probes installed in the Orange

TABLE I. Download Statistics in Bordeaux.

	Number of distinct files	Number of downloads	Total file volume
Day 7			
Files	21 354	29 604	2 278.1 GB
More than twice	1 237	7 197 (24.3 %)	638.1 GB
Only once		17 825 (60.2 %)	1 182.0 GB
First week			
Files	93 649	159 486	5 782 GB
More than twice	9 369	63 379 (39.7 %)	2 598.9 GB
Only once		72 451 (45.4 %)	2 249.7 GB
Two weeks			
Files	202 672	453 345	9 442.2 GB
More than twice	25 981	249 243 (54.9 %)	4 730.7 GB
Only once		149 280 (32.9 %)	3 315.5 GB

[1] A possible work-around would be to consider caching at chunk level. We cannot guarantee, however, that all users are using the same fixed set of chunks for a single video file (e.g. due to on the fly re-encoding to different bit rates or format depending on the user conditions and preferences), and thus making this possibility uncertain.

TABLE II. DOWNLOAD STATISTICS IN MONITORED ADSL AREAS.

	Number	Number of downloads	Volume
Day 7			
Files	104 548	223 422	6 493.9 GB
More than twice	9 420	115 516 (51.7 %)	2 757.9 GB
Only once		82 350 (36.8 %)	2 634.6 GB
First week			
Files	340 702	836 236	12 484.6 GB
More than twice	43 480	492 482 (58.9 %)	6 124.7 GB
Only once		250 690 (29.9 %)	4 463.6 GB
Two weeks			
Files	865 681	2 838 514	22 632.5 GB
More than twice	145 724	1 992 348 (70.2 %)	11 916.9 GB
Only once		593 721 (20.9 %)	7 483.1 GB

IP backbone network. We qualitatively observe the same phenomena as those mentioned above in Bordeaux regarding file and volume statistics. Note that volumes generated within two weeks by those files viewed more than twice become huge (recall that only 25 % of all ADSL areas are supervised by probes).

C. File popularity

In order to better characterize file requests distribution, we now compute file popularity curves. We have computed on a day-by-day basis the popularity of files viewed in Bordeaux during the first observation week. Specifically, we have recorded the (YouTube) ID of files viewed within one day in Bordeaux and their associated number of occurrences. We have then classified files in decreasing occurrence order and computed their popularity index defined as the ratio of the number of times the file is viewed to the total number of downloads. This procedure yields the popularity curve of files viewed in Bordeaux, which is depicted in Figure 1(a). We have displayed the popularity of those files which are seen a sufficiently large number of times. There is a long tail of files which are rarely viewed, which yields an erratic popularity curve; this point will be further discussed below.

It is remarkable that the popularity curves are similar from one day to another, even if the files are not the same[2]. We have applied the same procedure for data collected during the first day of three consecutive weeks (Figure 1(b)) and during the two weeks by aggregating the data for the first week, the second week and both weeks (Figure 1(c)). It clearly appears from these figures that the popularity curves are similar to those computed within one day and can be approximated by the same function.

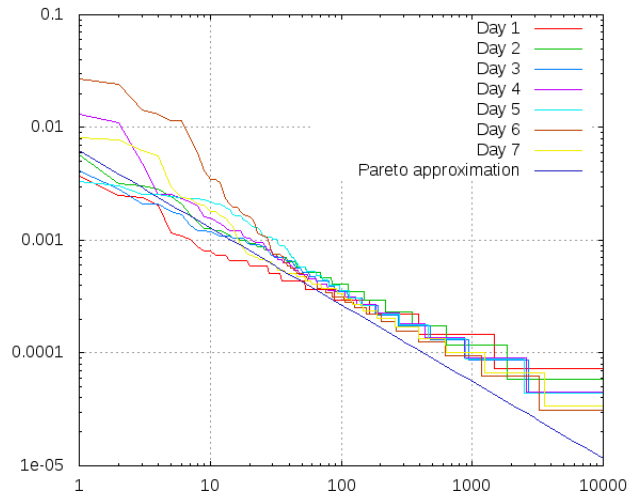
The same procedure has been applied to data collected in Lyon, in Paris and by all probes and we have observed the same phenomena, namely that popularity curves are similar from day to day and over larger aggregation periods (week by week and for the two weeks of observation).

In Figure 1, we approximated the popularity curves by means of a truncated Pareto (or Zipf) curve of the form

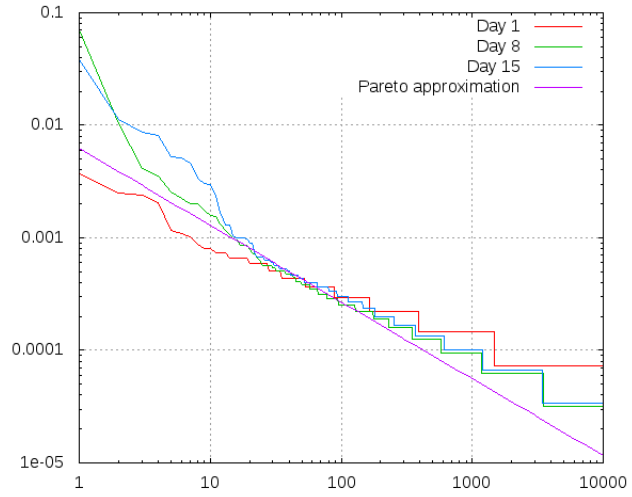
$$q_r = \mathbb{1}_{\{r_{\min} \le r \le r_{\max}\}} \frac{A}{r^{\alpha}}, \qquad (1)$$

where index r denotes the file popularity rank. The value of parameters A and α as well as the range $[r_{\min}, r_{\max}]$ may

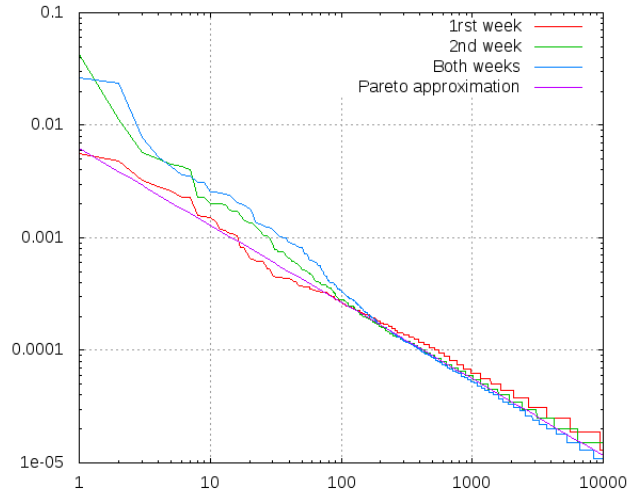
[2]This point related to the behavior of users when browsing YouTube requires further investigations, but this aspect is out of the scope of the present paper.



(a) One week.



(b) The first days of three consecutive weeks.



(c) Aggregation over the first and second weeks and over the two weeks.

Fig. 1. Popularity of YouTube video files in Bordeaux.

obviously change, depending on the duration of measurement windows and the location where measurements are performed. When $\alpha < 1$, the range $[r_{\min}, r_{\max}]$ is necessarily finite. By using the Marquardt-Levenberg nonlinear least-squares regression algorithm (implemented in Gnuplot), we found that the file popularity curve can be well approximated by a truncated Pareto function in the range $[10, 1000]$ and with coefficients

$$A = 0.006217, \quad \alpha = 0.68198.$$

The same approximation holds also for the file popularity distribution observed in Lyon, in Paris and by all probes.

We now emphasize the fact that the Pareto approximation holds only for a small proportion of the file population. In Figure 2, we have displayed the complete popularity curve for files viewed in Bordeaux during the two weeks. There were 453 345 downloads; the latter can be decomposed into the three following segments:

- **Heavy hitters:** Files with rank less than $r_{\min}$, which are massively viewed by end-users. Their popularity is higher than that estimated by the Pareto approximation. Their number is small but they represent a huge amount of transferred data (the 10 most popular files give rise to 32.6 TB of traffic, 23.3 % of global traffic in Bordeaux). This unbalanced contribution of files has been already observed in [11] and corresponds more or less to the well-known Pareto rule (20 % of a population owns 80 % of the resources) but for YouTube files, this rule is much sharper (a tiny proportion of files contributes to the majority of volume);
- **Pareto class:** Beyond heavy hitters, there is a subpopulation of files which are viewed a significant number of times and for which a Pareto approximation (in the sense of Equation (1)) holds. The need for this approximation may appear artificial at first glance but, as we shall see in next section, such an approximation can be used to estimate cache performance;
- **Noise:** As shown in Table I, 33 % of files are viewed only once, thus showing a high volatility of YouTube requests in a commercial environment. When considering the problem of caching, these files may appear as a noise; in fact, they enter a cache if no filtering is performed and a standard LRU management policy is implemented for replacing content in the cache[3].

III. Mathematical model

In view of the empirical observations made in the previous section, we develop in the following a mathematical model to estimate the file and byte hit ratios of a cache fed with a file arrival process with two components, namely regular files with a Zipf popularity distribution and noise files irrelevant for caching.

[3]It is proved in [12] that filtering requests improves the hit ratio by allowing files to enter the cache only after a fixed number of requests, but degrades the byte hit ratio since heavy hitters can enter the cache only after a fixed number of times. This observation incites us not to filter out requests since the byte hit ratio is actually the most relevant metric to estimate the gain in terms of bandwidth savings.

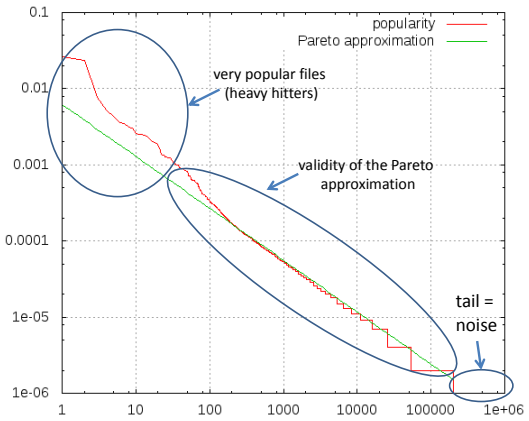


Fig. 2. Complete file popularity curve of YouTube video files in Bordeaux.

A. Model description

Consider a cache server whose replacement policy is LRU [13], [14]. The objects contained in that cache may come from either:

- a "persistent" class $\mathcal{P}$: Requests addressed to objects within $\mathcal{P}$ build up a stationary Poisson process in time with rate Λ (requests/sec.). The request rate for object numbered r in class $\mathcal{P}$ is therefore Λq_r, where q_r is the stationary popularity of object $r \in \mathcal{P}$ (objects are ranked in decreasing order of popularity), with $\sum_{r\in\mathcal{P}} q_r = 1$;
- a "non-persistent" class $\mathcal{N}$ considered as a noise: All requests addressed to objects within $\mathcal{N}$ form a Poisson process with rate $\Lambda_{\mathcal{N}}$; any object $n \in \mathcal{N}$ is assumed to be requested just once within time interval $[0, +\infty[$.

Following the discussion of section II-C, the number of heavy hitters is small enough as to consider it as negligible with respect to the cache capacity. Their hit ratio is close to 1 (see section IV) and their impact on the hit ratio of objects in class $\mathcal{P}$ is neglected.

First assume that the cache capacity C is measured as a number of objects (or files). We then denote by M_r the probability ("conditional object miss probability") that a request for object $r \in \mathcal{P}$ is not satisfied; we also write

$$M_{\mathcal{P}} = \sum_{r\in\mathcal{P}} q_r M_r \tag{2}$$

for the average conditional object miss probability, when averaged over all objects of class $\mathcal{P}$. Recall that by definition any request for an object $n \in \mathcal{N}$ is not satisfied and has therefore miss probability equal to 1.

The popularity distribution attached to $\mathcal{P}$ will be assumed to be Zipf with exponent parameter $\alpha \in]0, 1[$, that is,

$$q_r = \frac{A_N}{r^\alpha}, \quad 1 \le r \le N, \tag{3}$$

where N is the total number of objects ("catalog size") in class $\mathcal{P}$ and A_N is the associated normalizing constant.

We now evaluate the impact of the noisy class $\mathcal{N}$, characterized by parameter $\Lambda_{\mathcal{N}}$, on the performance of the persistent class $\mathcal{P}$. As detailed below, such an evaluation mainly relies on the so-called "Che approximation" [15], [16] for deriving estimates of miss probabilities for large cache capacity C.

B. Object miss probabilities

Let $Q(t)$ denote the number of *distinct* objects requested within time interval $[0,t[$; if $A_r(t)$ (resp. $B_n(t)$) is the number of requests for object $r \in \mathcal{P}$ (resp. object $n \in \mathcal{N}$), we consequently have

$$Q(t) = \sum_{r\geq 1} \mathbf{1}_{\{A_r(t)\geq 1\}} + \sum_{n\geq 1} \mathbf{1}_{\{B_n(t)\geq 1\}},$$

where $\mathbf{1}_A$ is the indicator function of the set A; by definition of Poisson processes introduced above and the fact that all objects in class $\mathcal{N}$ are distinct, the mean $q(t) = \mathbb{E}(Q(t))$ equals

$$q(t) = \sum_{r\geq 1}(1 - e^{-\Lambda q_r t}) + \Lambda_{\mathcal{N}} t. \tag{4}$$

Define the critical time T as the time necessary to totally renew the cache population, so that a miss event occurs for an object with class r if the time interval between two consecutive requests for that object is larger than T. Following the Che approximation, T is assumed to be a constant (depending on C) which verifies equation

$$q(T) = C. \tag{5}$$

Note, in particular, that the critical time T increases with the capacity C. As requests for object $r \in \mathcal{P}$ form a Poisson process whose inter-arrivals are exponentially distributed with parameter Λq_r, the conditional object miss probabilities M_r, $r \in \mathcal{P}$ is then evaluated by

$$M_r \sim e^{-\Lambda q_r T} \tag{6}$$

for large capacity C.

Within time interval $[0,t[$, the mean number of objects requested from the cache for object $r \in \mathcal{P}$ (resp. for any object $n \in \mathcal{N}$) is $\Lambda q_r t$ (resp. $\Lambda_{\mathcal{N}} t$), while the mean number of unsatisfied requests for object $r \in \mathcal{P}$ (resp. for any object $n \in \mathcal{N}$) is $\Lambda q_r M_r t$ (resp. $\Lambda_{\mathcal{N}} t$ as well); the law of large numbers for $t \uparrow +\infty$ then entails that the *average miss probability* m can be expressed as

$$m = \frac{\displaystyle\sum_{r\geq 1} \Lambda q_r M_r + \Lambda_{\mathcal{N}}}{\displaystyle\sum_{r\geq 1} \Lambda q_r + \Lambda_{\mathcal{N}}} = \frac{M_{\mathcal{P}} + \dfrac{\Lambda_{\mathcal{N}}}{\Lambda}}{1 + \dfrac{\Lambda_{\mathcal{N}}}{\Lambda}}. \tag{7}$$

We now address the evaluation of miss probabilities for the popularity distribution (3) of class $\mathcal{P}$. We further assume that C and N both scale as

$$C = \delta N \tag{8}$$

for some positive constant $\delta < 1$. Recall [17] that the incomplete Gamma function is defined by

$$\Gamma(s;x) = \int_x^{+\infty} e^{-u} u^{s-1} \mathrm{d}u \tag{9}$$

for $x \in \mathbb{R}$, $s > 0$, and that it can be extended to non integer negative values of s through the recursion

$$\Gamma(s-1;x) = \frac{\Gamma(s;x) - x^{s-1}e^{-x}}{s-1}$$

for all $x \in \mathbb{R}$.

Proposition III.1: For $0 < \alpha < 1$ and large C so that scaling condition (8) is fulfilled, the conditional miss probability M_r is estimated by

$$M_r \sim \exp\left(-\frac{q_r}{\delta}\frac{\Theta}{1-\alpha}C\right) \tag{10}$$

for given $r \in \mathcal{P}$, where Θ is the unique positive solution to equation

$$\frac{\Theta^{\frac{1}{\alpha}}}{\alpha}\Gamma\left(-\frac{1}{\alpha};\Theta\right) = 1 - \delta + \frac{\Lambda_{\mathcal{N}}}{\Lambda}\frac{\Theta}{1-\alpha}. \tag{11}$$

The average miss probability $M_{\mathcal{P}}$ is estimated by

$$M_{\mathcal{P}} \sim \frac{1-\alpha}{\alpha}\Gamma\left(1-\frac{1}{\alpha};\Theta\right)\Theta^{\frac{1}{\alpha}-1} \tag{12}$$

in identical asymptotic conditions.

The proof of Proposition III.1 is detailed in Appendix B. Solution Θ depends on ratio $\Lambda_{\mathcal{N}}/\Lambda$ through implicit equation (11); as expected, the latter coincides with that derived in [18, Theorem 2] in the case when the noise traffic is absent, that is, $\Lambda_{\mathcal{N}} = 0$ and parameter k in [18] fixed to 1. We note, in particular, that miss probability M_r given by (10) increases when $\Lambda_{\mathcal{N}}$ increases.

C. Byte miss probability

Measuring the cache capacity as a number of objects implicitly assumes that all objects have identical size in bytes. If the cache capacity is rather measured in bytes, say $\widetilde{C}$, the definitions of Section III-A for miss probabilities can be adapted accordingly.

Let V_r be the size (or volume) in bytes of object $r \in \mathcal{P}$; let similarly $V_{\mathcal{N}}$ be the mean size in bytes of objects $n \in \mathcal{N}$. If $\widetilde{M}_r$ then denotes the probability ("conditional object miss probability") that a request for object $r \in \mathcal{P}$ is not satisfied, we write

$$\widetilde{M}_{\mathcal{P}} = \sum_{r\in\mathcal{P}} q_r \widetilde{M}_r \tag{13}$$

for the unconditional object miss probability, when averaged over all objects of class $\mathcal{P}$.

Let $\widetilde{Q}(t)$ denote the number of bytes generated by *distinct* objects requested within time interval $[0,t[$; if $A_r(t)$ (resp. $B_n(t)$) is the number of requests for object $r \in \mathcal{P}$ (resp. object $n \in \mathcal{N}$) to the cache, we have

$$\widetilde{Q}(t) = \sum_{r\geq 1} V_r \mathbf{1}_{\{A_r(t)\geq 1\}} + \sum_{n\geq 1} V_{\mathcal{N}} \mathbf{1}_{\{B_n(t)\geq 1\}};$$

by arguments similar to that invoked in III-B, we deduce that the mean $\widetilde{q}(t) = \mathbb{E}(\widetilde{Q}(t))$ equals

$$\widetilde{q}(t) = \sum_{r\geq 1} V_r(1 - e^{-\Lambda q_r t}) + V_{\mathcal{N}}\Lambda_{\mathcal{N}} t. \tag{14}$$

Define the "critical time" $\widetilde{T}$ (depending on $\widetilde{C}$) as the time necessary to totally renew the cache population when counted in bytes. Following the Che approximation, $\widetilde{T}$ verifies the equation

$$\widetilde{q}(\widetilde{T}) = \widetilde{C}. \tag{15}$$

For large capacity $\widetilde{C}$, the conditional object miss probabilities $\widetilde{M}_r$, $r \in \mathcal{P}$ is then evaluated by

$$\widetilde{M}_r \sim e^{-\Lambda q_r \widetilde{T}}. \tag{16}$$

Within interval $[0, t[$, the mean number of bytes requested from the cache for object $r \in \mathcal{P}$ (resp. for any object $n \in \mathcal{N}$) is $\Lambda q_r V_r t$ (resp. $\Lambda_{\mathcal{N}} V_{\mathcal{N}} t$), while the mean number of bytes satisfied for requests from object $r \in \mathcal{P}$ (resp. for requests from object $n \in \mathcal{N}$) is $\Lambda q_r V_r \widetilde{M}_r t$ (resp. $\Lambda_{\mathcal{N}} V_{\mathcal{N}} t$ as well); it then follows that the *average byte miss probability* $\widetilde{m}$ can be expressed as

$$\widetilde{m} = \frac{\displaystyle\sum_{r \geq 1} \Lambda q_r V_r \widetilde{M}_r + \Lambda_{\mathcal{N}} V_{\mathcal{N}}}{\displaystyle\sum_{r \geq 1} \Lambda q_r V_r + \Lambda_{\mathcal{N}} V_{\mathcal{N}}}. \tag{17}$$

Let us further assume that $V_r = V$ does not depend on r and that the cache size $\widetilde{C}$ and the total volume of the persistent class $\mathcal{P}$ scales similarly. Define $\widetilde{\delta} = \widetilde{C}/(NV)$; let also $\mathcal{V}$ (resp. $\mathcal{V}_{\mathcal{N}}$) denote the total downloaded volume of class $\mathcal{P}$ (resp. of class $\mathcal{N}$). Note that the ratio $\mathcal{V}_{\mathcal{N}}/\mathcal{V}$ is equal to $(\Lambda_{\mathcal{N}} V_{\mathcal{N}}) / (\Lambda V)$. The conditional miss probability $\widetilde{M}_r$ can then be computed using the following proposition:

Proposition III.2: For $0 \leq \alpha < 1$ and large $\widetilde{C}$, the conditional miss probability $\widetilde{M}_r$ is estimated under the above assumptions by

$$\widetilde{M}_r \sim \exp\left(-\frac{q_r}{\widetilde{\delta}} \frac{\widetilde{\Theta}}{1-\alpha} \widetilde{C} \right) \tag{18}$$

for given $r \in \mathcal{P}$, where $\widetilde{\Theta}$ is the positive solution to equation

$$\frac{\widetilde{\Theta}^{\frac{1}{\alpha}}}{\alpha} \Gamma\left(-\frac{1}{\alpha}; \widetilde{\Theta} \right) = 1 - \widetilde{\delta} + \frac{\mathcal{V}_{\mathcal{N}}}{\mathcal{V}} \frac{\widetilde{\Theta}}{1-\alpha}. \tag{19}$$

The average miss probability $\widetilde{M}_{\mathcal{P}}$ for the Pareto class is estimated by

$$\widetilde{M}_{\mathcal{P}} \sim \frac{1-\alpha}{\alpha} \Gamma\left(1 - \frac{1}{\alpha}; \widetilde{\Theta} \right) \widetilde{\Theta}^{\frac{1}{\alpha}-1}. \tag{20}$$

Proof: Invoking the Che approximation and equating all file sizes V_r to their joint value V, relations (14) and (15) together yield

$$\sum_{r \geq 1} (1 - e^{-\Lambda q_r \widetilde{T}}) + \frac{V_{\mathcal{N}}}{V} \Lambda_{\mathcal{N}} \widetilde{T} = \frac{\widetilde{C}}{V}.$$

The calculation of miss rates in case of variable file sizes is therefore similar to the application of Proposition III.1 when changing $\Lambda_{\mathcal{N}}/\Lambda$ and C to $\mathcal{V}_{\mathcal{N}}/\mathcal{V}$ and $\widetilde{C}/V$, respectively (see Equation (25) in Appendix B). ■

To conclude this section, let us note that cache capacities C and $\widetilde{C}$ can be related by setting the latter to equal the mean cache occupancy expressed in bytes, that is,

$$\widetilde{C} = \sum_{r \geq 1} V_r (1 - M_r) + V_{\mathcal{N}} \Lambda_{\mathcal{N}} T; \tag{21}$$

in fact, object $r \in \mathcal{P}$ is present in cache with probability $1 - M_r$ while any object $n \in \mathcal{N}$ has an expected sojourn time in cache T and the total occupancy of class $\mathcal{N}$ in cache is $\Lambda_{\mathcal{N}} V_{\mathcal{N}} \times T$ by Little's law. Conversely, C can be expressed in terms of the byte capacity $\widetilde{C}$ as

$$C = \sum_{r \geq 1} (1 - \widetilde{M}_r) + \Lambda_{\mathcal{N}} \widetilde{T} \tag{22}$$

by means of similar arguments.

IV. Trace driven simulation

In this section, we compare the theoretical results obtained above against trace driven simulation results, and derive simple cache dimensioning rules.

A. Experimental results

To perform trace driven simulation, we specifically consider a 1 TB cache running a standard LRU replacement policy; we use the YouTube traffic traces and replay the sequence of file requests. Each file is associated with its maximal size, i.e., the maximum observed volume for a given YouTube ID. This is clearly an upper bound for the volume really transmitted (see the discussion in section II).

By using the data for Bordeaux, Lyon and Paris during the two observation weeks, we have computed the various file and byte hit ratios for the above LRU cache, as shown in Table III. While the global file hit ratio is rather small, notably because of "noise" files which are very numerous but completely irrelevant for caching, the byte hit ratio is quite high because the heavy hitter files are very large. Moreover, the volume of content delivered by the cache is large, even if the cache capacity is rather small. For instance, the total downloaded volume in Bordeaux is 140.9 TB and the cache is able to deliver more than 105 TB.

TABLE III. Request and byte hit ratio for a 1 TB cache in Bordeaux, Lyon, and Paris.

Location	Hit ratio	Byte hit ratio	Downloaded volume
Bordeaux	33.5 %	74.65 %	140.9 TB
Lyon	34.2 %	74.61 %	161.3 TB
Paris	34.6 %	77.97 %	240.4 TB

TABLE IV. Hit ratio of heavy hitters in Bordeaux.

hit ratio	# served by cache	# requests	volume
0.999916	11943	11944	1.2 GB
0.999812	10651	10653	400 MB
0.999717	3534	3535	400 MB
0.995773	2356	2366	1.9 GB
0.999493	1971	1972	700 MB
0.999389	1635	1636	1.4 GB
0.997503	1598	1602	500 MB
0.999289	1405	1406	1.4 GB
0.992137	1388	1399	700 MB
0.998285	1164	1166	600 MB

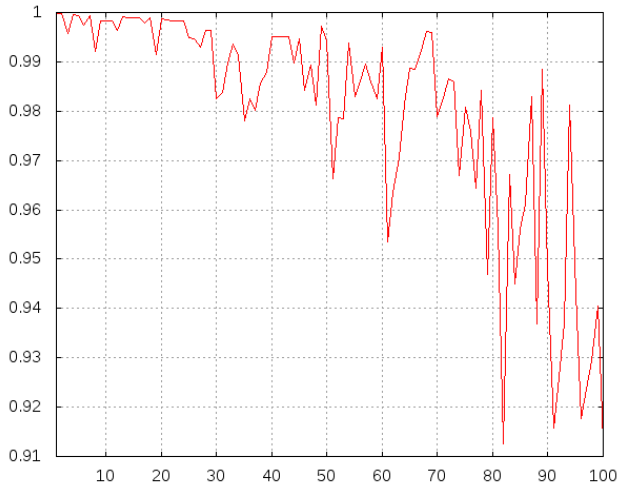


Fig. 3. Hit ratio of the 100 most popular file in Bordeaux with a LRU cache of 1 TB.

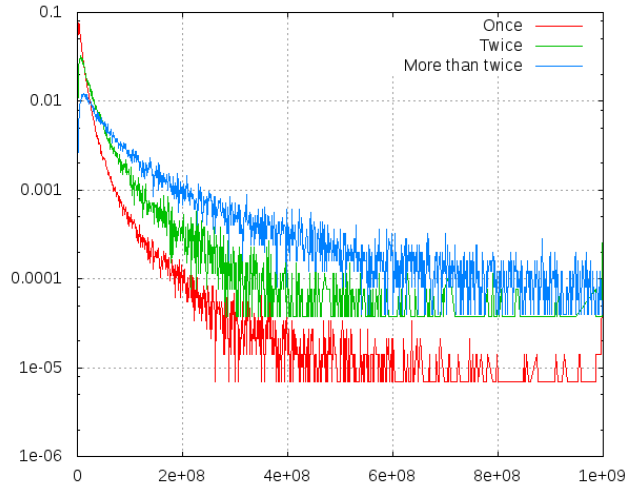


Fig. 4. Probability density functions of file sizes viewed once, twice and more than twice.

Table IV reports the hit ratios, the numbers of times a file has been downloaded from the cache, the numbers of requests together with the volumes of the 10 most popular files (heavy hitters) in Bordeaux. The volume of data generated by these files amounts to 32.6 TB (i.e., about 23.3 % of global traffic) but their cumulative volume is 9.2 GB, which is much less than the cache capacity equal to 1 TB. As in Section III, we can ignore the heavy hitters and just consider the files in the Pareto class.

In Figure 3, we have plotted the hit ratio for the 100 top files ordered in decreasing order of popularity. We clearly observe that the hit ratio of those most popular files is high, indicating that the cache, even with a rather limited storage capacity, is able to store heavy hitters. We also note that the hit ratio rapidly decreases with the popularity.

B. Comparison with theoretical results

In order to apply the results of Section III, we first compute the probability density function of the file sizes viewed in Bordeaux (see Figure 4). We can observe that the densities depend on the popularity; in particular, the mean file size is 22.2 MB for files viewed once, 51 MB for those viewed twice and 182 MB for those viewed more than twice. The mean file size for the $N = 53\,424$ files for which the Pareto approximation roughly holds is equal to $V = 114.2$ MB; the mean file size for the other files (noisy class) equals $V_{\mathcal{N}} = 22.2$ MB. We nevertheless shall apply the results of Section III-C to compare the approximation obtained in that section against experimental results.

From Tables I and III, we measure

$$\mathcal{V}_{\mathcal{N}} = 3.3\text{TB} \ll \mathcal{V} = 105\text{TB}.$$

(Note that $\mathcal{V}$ is the total volume of downloads equal to 140.9 TB minus the sum of the volumes of noise and heavy hitter downloads equal to 32.6 TB.) The number of Pareto class files is $N = 53\,382$.

Following Proposition III.2, the miss rate $\widetilde{M}_r$ for the rth most popular file is consequently given by

$$\widetilde{M}_r \sim \exp\left(-\frac{q_r}{\widetilde{\delta}}\frac{\widetilde{\Theta}}{1-\alpha}\frac{\widetilde{C}}{V}\right) \sim \exp\left(-\left(\frac{N}{r}\right)^{\alpha}\widetilde{\Theta}\right)$$

for large N with $\widetilde{\delta} = \widetilde{C}/(NV) = O(1)$, where $\widetilde{\Theta}$ is the solution to equation

$$\frac{\widetilde{\Theta}^{\frac{1}{\alpha}}}{\alpha}\Gamma\left(-\frac{1}{\alpha};\widetilde{\Theta}\right) = 1 - \widetilde{\delta} + \frac{\mathcal{V}_{\mathcal{N}}}{\mathcal{V}}\frac{\widetilde{\Theta}}{1-\alpha}.$$

In the case presently considered ($\mathcal{V}_{\mathcal{N}} = 3.3$ TB and $\mathcal{V} = 105$ TB), the value of $\widetilde{\Theta}$ solving Equation (19) is equal to 0.0725. By using that value of $\widetilde{\Theta}$, we obtain an average miss rate for the Pareto class equal to 57.3 %. On the other hand, trace driven simulations give 114 026 hits among 266 386 requests for the Pareto class, that is, an average miss rate equal to 57.2 %.

The fact that these numerical estimates are so close hides two model limitations which balance each other. More specifically, the model does not take into account the variation of file sizes with the popularity, which leads to an overestimation of the hit ratio, as argued in Appendix A. On the other hand, some files are viewed in bursts for a limited period of time and then disappear; they thus have a hit ratio close to 1, whereas the model assumes that requests are randomly spread over the observation period, and therefore predicts a lower hit ratio than the actual one for these files.

Figure 5 displays the hit ratios of files ordered by their popularity, and illustrates the two above-mentioned effects. It clearly appears that these hit rates greatly vary, up to a ratio close to 1 for those non-popular files which are requested in bursts. We finally observe that a significant proportion of the measured hit ratios lies below the approximation curve. This may again be explained by the above-mentioned effect of the file size variation. Estimate (18), nevertheless, provides the global behavior for the file hit ratios and allows us to evaluate the average hit ratio of Youtube files.

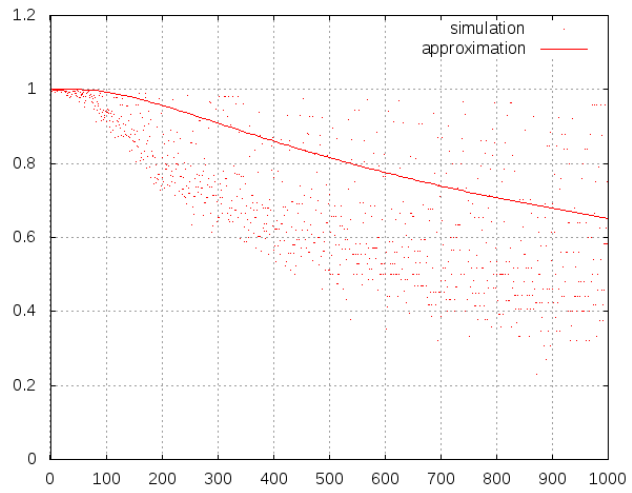


Fig. 5. Experimental hit ratios of files in Bordeaux ordered in decreasing order of popularity and analytical approximation.

V. Conclusion

By using data from the Orange IP backbone network in France, we have reported statistics for YouTube traffic in a commercial environment. While YouTube traffic is highly volatile with a large number of files viewed only once, there is a subset of files which are massively viewed by end users and consequently worth caching in the network. By caching those files, significant resource savings can be achieved; in fact, a cache located at the Bordeaux Point of Presence (PoP) of the operator with a limited storage capacity of 1 TB achieves a byte hit ratio of about 75 %.

On the basis of the above observations, we have developed an analytical model which extends the ones considered so far in the literature in the sense that a "noisy" traffic component has been here included. Besides the noisy component, the popularity curve of files can be approximated by a truncated Pareto function with shape parameter $\alpha < 1$; this enables us to obtain estimates for file hit ratios.

Our model gives a good estimate for the average hit ratios, although it has some inherent limitations. On the one hand, it does not account for the correlation between file size and file popularity. On the other hand, it ignores the correlated dynamics of file requests. These two aspects, which compensate each other, will be investigated in forthcoming studies.

Appendix

A. Impact of file size variation on the hit ratio

Denote by X (resp. $\widetilde{X}$) the (random) number of files present in a cache with capacity C files (resp. with capacity $\widetilde{C}$ bytes). In our model, we assume that all files (except noise files, which we ignore for simplicity in this discussion) have an identical size V in bytes; the associated average hit ratio m thus corresponds to that of a cache with capacity $C = \widetilde{C}/V$ files, and we have $X = C = \widetilde{C}/V$.

On the other hand, the simulation uses actual traces, where file r has a specific size V_r. Denoting by $\widetilde{V}$ the (random) mean size of a file stored in the cache, we have by definition $\widetilde{V} = \widetilde{C}/\widetilde{X}$, hence $\mathbb{E}[\widetilde{X}] = \widetilde{C}\mathbb{E}[1/\widetilde{V}] \approx \widetilde{C}/\mathbb{E}[\widetilde{V}]$, where the expectation is taken w.r.t. the cache population. But, if the most popular files are in average larger in size than the other files, we have $\mathbb{E}[\widetilde{V}] > V$ and consequently $\mathbb{E}[\widetilde{X}] < X$. This amounts to claim that the hit ratio estimated by our model can be considered as an upper bound of the actual hit ratio measured by the simulation.

B. Proof of Proposition III.1

We here detail the proof of Proposition III.1. We first solve equation (5) for T, that is, $q(T) = C$ with large C. Estimating the finite sum in (4) by an integral, we obtain

$$\begin{aligned} q(T) &= \sum_{r\geq 1}(1-e^{-\Lambda q_r T}) + \Lambda_{\mathcal{N}} T \\ &\sim \int_1^N \left[1 - \exp\left(-\frac{\Lambda A_N T}{x^\alpha}\right)\right] \mathrm{d}x + \Lambda_{\mathcal{N}} T \\ &= \frac{(\Lambda A_N T)^{\frac{1}{\alpha}}}{\alpha} \int_{\frac{\Lambda A_N T}{N^\alpha}}^{\Lambda A_N T} (1-e^{-u}) \frac{\mathrm{d}u}{u^{1+\frac{1}{\alpha}}} + \Lambda_{\mathcal{N}} T. \end{aligned} \quad (23)$$

As

$$\frac{1}{A_N} = \sum_{r=1}^{N} r^{-\alpha} \sim \frac{N^{1-\alpha}}{1-\alpha}$$

tends to infinity for increasing N, the lower and upper bounds in the latter integral are evaluated as

$$\frac{\Lambda A_N T}{N^\alpha} \sim (1-\alpha)\frac{\Lambda T}{N}, \quad \Lambda A_N T \sim (1-\alpha)\frac{\Lambda T}{N^{1-\alpha}}.$$

Define $\Theta = (1-\alpha)\Lambda T/N$; estimate (23) for $q(T)$ then reads

$$\begin{aligned} q(T) &\sim \Theta^{\frac{1}{\alpha}} \frac{N}{\alpha} \int_\Theta^{\Theta N^\alpha} (1-e^{-u}) \frac{\mathrm{d}u}{u^{1+\frac{1}{\alpha}}} + \Lambda_{\mathcal{N}} T \\ &\sim \frac{\Theta^{\frac{1}{\alpha}}}{\alpha\delta} C \int_\Theta^{+\infty} (1-e^{-u}) \frac{\mathrm{d}u}{u^{1+\frac{1}{\alpha}}} + \Lambda_{\mathcal{N}} T \end{aligned} \quad (24)$$

where $C = \delta N$ on account of scaling condition (8).

It follows that (5) is verified when $T = O(N) = O(C)$, so that $\Theta = O(1)$ and $q(T) = O(C)$ by (24);the last integral in (24) further reduces to

$$\int_\Theta^{+\infty} (1-e^{-u}) \frac{\mathrm{d}u}{u^{1+\frac{1}{\alpha}}} = \alpha \left[\frac{1}{\Theta^{\frac{1}{\alpha}}} - \frac{1}{\alpha}\Gamma\left(-\frac{1}{\alpha};\Theta\right)\right].$$

From the above evaluation, equation (5) then reduces to

$$C \cdot \frac{\Theta^{\frac{1}{\alpha}}}{\alpha\delta}\left[\frac{\alpha}{\Theta^{\frac{1}{\alpha}}} - \Gamma\left(-\frac{1}{\alpha};\Theta\right)\right] + \frac{\Lambda_{\mathcal{N}}}{\Lambda(1-\alpha)}\frac{C}{\delta}\Theta = C \quad (25)$$

that is, equation (11).

We now show that equation (11) has actually a unique positive solution Θ. Let $r = 1/\alpha$ for short and define

$$f_r(\theta) = r\theta^r \Gamma(-r, \theta)$$

for $\theta > 0$ so that equation (11) reads

$$f_r(\Theta) = 1 - \delta + p\Theta \tag{26}$$

with $p = \Lambda_{\mathcal{N}}/\Lambda(1-\alpha)$. It is easily verified that function f_r is continuous and decreasing from $f_r(0) = 1$ to $f_r(+\infty) = 0$ (in fact, $\mathrm{d}f_r(\theta)/\mathrm{d}\theta = -r\theta^{r-1}\Gamma(1-r,\theta) < 0$ for any $\theta > 0$). In the (O, θ, ξ) positive quadrant, the curve $\xi = f_r(\theta)$ therefore meets the line $\xi = 1 - \delta + p\theta$ exactly once at some positive abscissa $\theta = \Theta > 0$, given $0 < \delta < 1$ and $p > 0$. Equation (26), or equivalently (11), has therefore a unique positive solution Θ. By general asymptotic (6), estimate (10) for M_r then follows.

Besides, definition (2) for $M_{\mathcal{P}}$ and the latter estimate for M_r give

$$M_{\mathcal{P}} = \sum_{r\geq 1} q_r M_r \sim \sum_{r=1}^{N} \frac{A_N}{r^\alpha} e^{-J_\alpha/r^\alpha} \sim A_N \int_1^N e^{-J_\alpha/x^\alpha} \frac{\mathrm{d}x}{x^\alpha}$$

where we have approximated the Riemann sum by the associated integral and with

$$J_\alpha = \Lambda A_N T = \frac{\Theta C}{\delta(1-\alpha)} A_N \sim \frac{\Theta C^\alpha}{\delta^\alpha}.$$

The variable change $u = J_\alpha/x^\alpha$ in the latter integral and the fact that $J_\alpha \uparrow +\infty$ when $C \uparrow +\infty$ provide final estimate (12) for miss probability $M_{\mathcal{P}}$ ■

We depict in Fig.6 the variations of function $f_{1/\alpha}$ and the corresponding abscissas Θ and η (the latter corresponding to $\Lambda_{\mathcal{N}} = 0$); the graph also suggests that these abscissas are decreasing functions of parameter $\alpha < 1$.

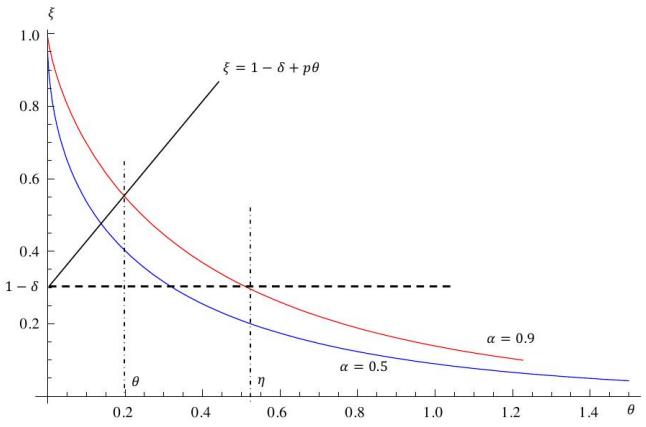


Fig. 6. *Graph of function $f_{1/\alpha}$ for Zipf parameters $\alpha = 0.5$ (blue) and $\alpha = 0.9$ (red). For $\alpha = 0.9$, intersections with horizontal line $\xi = 1 - \delta$ (case $\Lambda_{\mathcal{N}} = 0$) and $\xi = 1 - \delta + p\theta$ (case $\Lambda_{\mathcal{N}} \neq 0$) are indicated.*

ACKNOWLEDGMENT

The authors would like to thank their colleague Thierry Houdoin for providing them with YouTube traffic traces. This work has partially been supported by the VIPEER (ANR-09-VERSO-014) and CONNECT (ANR-10-VERSO-001) projects.

REFERENCES

[1] "Global Internet phenomena report," Sandvine, Tech. Rep., 2011.

[2] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On dominant characteristics of residential broadband Internet traffic," in *Proc. IMC'09*, 2009, pp. 90 – 102.

[3] Y. Carlinet, B. Kauffmann, P. Olivier, and A. Simonian, "Trace-based analysis for caching multimedia services," Orange Labs, Tech. Rep., 2011.

[4] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, IMC'07*, 2007, pp. 15–28.

[5] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of Youtube traffic at a campus network - Measurements, models, and implications," *Computer networks*, vol. 53, pp. 501 – 514, 2009.

[6] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 17, no. 5, 2009.

[7] Y. Carlinet, T. D. Huynh, B. Kauffmann, F. Mathieu, L. Noirie, and S. Tixeuil, "Four months in dailymotion: Dissecting user video requests," *International Workshop on TRaffic Analysis and Classification (TRAC)*, Aug. 2012.

[8] A. Rao, Y. Lim, C. Barakat, A. Legout, D. Towsley, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proc. ACM CoNext 2011*, Tokyo, Japan, December 2011.

[9] V. Adhikari, S. Janin, and Z. Zhang, "Youtube traffic deynamics and its interplay with a tier-1 ISP: An ISP perspective," in *Proc. IMC'10*, 2010, pp. 431–443.

[10] C. Labovitz, S. Iekel-Johnson, J. Oberheide, and F. Jahanian, "Internet inter-domain-traffic," in *Proc. Sigcomm'10*, New Dehli, India, 2010.

[11] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, IMC'07*, 2007, pp. 1 –14.

[12] F. Guillemin, T. Houdoin, and S. Moteau, "Volatility of YouTube content in Orange networks and consequences," in *Proc. ICC'13*, Budapest, June 2013.

[13] P. R. Jelenkovic, "Approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities," *Annals of Applied Probability*, vol. 9, no. 2, pp. 430 – 464, 1999.

[14] P. R. Jelenkovic and X. Kang, "Characterising the miss sequence of the lru cache," in *ACM SIGMETRICS, Performance Evaluation Review, Vol.36, 2, pp. 119 – 121*, 2008.

[15] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: modeling, design and experimental results," *IEEE JSAC*, vol. 20, no. 7, pp. 1305 – 1314, 2002.

[16] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *IEEE NOMEN 2012, Workshop on Emerging Design Choices in Name-Oriented Networking*, Orlando, March 2012.

[17] M. Abramowitz and I. Stegun, "Handbook of Mathematical Functions with formulas, graphs and mathematical tables," 1972.

[18] P. R. Jelenkovic and A. Radovanovic, "The persistent-access-caching algorithm," *Random Structures and Algorithms*, vol. 33, no. 2, pp. 219 – 251, 2008.