

I Tube, YouTube, P2PTube: Assessing ISP Benefits of Peer-assisted Caching of YouTube Content

Yann Nicolas, Daniel Wolff
Ecole Polytechnique, France
Email: first.last@polytechnique.edu

Dario Rossi
Telecom ParisTech, France
Email: dario.rossi@enst.fr

Alessandro Finamore
Politecnico di Torino, Italy
Email: finamore@tlc.polito.it

I. INTRODUCTION

The last few years have seen an explosion of video on demand traffic carried over the Internet infrastructure. While P2P applications have been proposed to carry VoD and TV content, they have so far encountered limited adoption except in Asian countries. Part of why this happens is explained with the fact that (i) the current asymmetric network infrastructure does not offer enough system capacity needed to let a fully P2P-VoD/TV to be self-sustainable, (ii) that the actual capacity at nominal peers is often smaller than the available one due to inefficiency in NAT punching[1], and (iii) the very same non-elastic nature of the service, that makes the system inherently less robust w.r.t elastic file-sharing to dynamic changes in the instantaneously available bandwidth. The other part of the story can be summarized with the success of CDN-managed services such as Netflix, Hulu and especially YouTube – according to [2], about 3 billion YouTube videos are viewed and 100's of thousand videos are uploaded *every day*, with independent research confirming YouTube to represent 20-30% of ISPs incoming traffic[3].

Yet, operators are sometimes struggling to handle the deluge of YouTube traffic. Recent, some tension arose between Google and ISPs as for the sheer traffic volume [4], [5]: as Google peers (in BGP sense) with ISPs, the latter struggle with its traffic for free. Yet, not only ISPs have no economic gain, but their pipes also carry advertisement that makes Google richer. As a results, frustrated ISPs either deteriorate user-performance [4] or block Google's advertisement as a stunning retorsion [5] for their loss of revenue.

II. SYSTEM-DESIGN

P2PTube offers a technical solution to this unfortunate, and avoidable, tension. We observe that set-top-boxes are ubiquitous, connected, permanently powered, have large hard-disks and are in control of the ISP. We therefore propose a simple yet effective system design, that we evaluate on trace-driven logs. Anticipating our results, we find that this tension can be easily relieved by half (on our dataset).

Being YouTube a HTTP-based service, a natural option would be to develop P2PTube as a browser plugin. While with the upcoming W3C standards P2P is likely to increasingly be present into browsers, we see several problems concerning (i) plugin penetration, (ii) security, (iii) inefficiency due to user churn, (iii) codebase deployment complexity (one per browser) (iv) management complexity (since the user base is globally distributed, we would be back to the relatively old issues of

proximity-aware peer selection in the P2P overlay). We argue that developing P2PTube as a LRU cache in the user set-top-box solves the penetration and security problems altogether (is in the control of the ISP), it is efficient (it eliminates churn as the box is always on, with possibly large hard-disk in reason of set-top-box and NAS functionalities), simple to deploy (single codebase, updates of firmware) and simple to manage (as management is delegated to each ISP).

With years of P2P reasearch, it would be unreasonable to reinvent the wheel: as such, we notice that YouTube videos are indexed with hashes, which recalls BitTorrent hashinfo – and all the successful ecosystem that BitTorrent has created over time. Hence, it make sense to follow its very same evolution: namely, small ISPs could start with a tracker-based infrastructure, while large ISPs would probably prefer distributed DHT-based indexing. As for the feasibility of a tracker-based solution, we point out that ThePirateBay succesfully tracks about 1 Million torrents (all of wich are not active at the same time), and that also recent work on caching of CDN content supports a similar tracker-based approach [6]. As for the advantages of a tracker-based solution, we note that peer selection could leverage IETF Application Layer Traffic Optimization (ALTO) functionalities, which would provide ISPs a simple management interface of P2PTube, and additionally provide a unified management interface in case of P2PTube federation among peering ISPs (in BGP sense).

Once the swarm is identified and peers are located, the content diffusion could go over BitTorrent. Yet, we point out that due to the spatial and temporal nature of video requests, chunking may not be necessary. Indeed, notice that (i) there is no user QoE benefit if the download rate exceeds the YouTube stream rate, (ii) videos are of short length 10MB on average[7] so that they would be constituted by few chunks anyway, (iii) emergence of large flash-crowd is less likely with respect to file-sharing (where peers stay longer due to the longer completion time), or P2P-TV and live-streaming (where arrivals as correlated with the beginning of the program). For the above reasons, a simpler approach where the content is downloaded by a single peers may be enough – as the average video rate is 250Kbps[3], P2PTubers will likely have spare uplink capacity even when streaming toward other peers.

III. PERFORMANCE EVALUATION

We obtain a 2-week trace from the authors of [3], corresponding to about 5000 (anonymized) users behind a DSLAM. As our logs only contain the request and the video size, but not the amount of video data actually downloaded at TCP level,

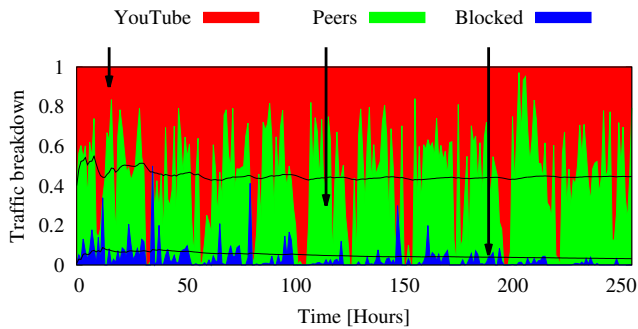


Fig. 1. Time evolution of P2PTube performance.

we cannot but assume the video to be fully viewed by users. Hence, despite videos are possibly interrupted earlier [3], for the above reason and to simplify management, we require the box to keep a complete copy of the video, that will be *entirely* streamed to another peer on demand.

Notice that we are not arguing against chunking [8] or parallel downloads [9], which are vital for fast epidemic dissemination of *large* contents to flash-crowd swarms. Rather, we believe that chunking is not needed for peer-assisted dissemination of *small* content (geometrically distributed size with average 10MB [7]), whose views are subject to loosely independent individual user arrivals. More specifically, we argue that in the case of YouTube, correlation of views of “viral” videos happens over time scales that are longer with respect to the duration of a single view. This fact, coupled to the relatively short video duration, makes it so that a peer can relatively easily download all the video from a single peer.

We first fix the uplink capacity to 500Kbps and the cache size to 100MB and evaluate caching performance in Fig. 1. The picture shows a time evolution of the percentage of videos served by YouTube CDN servers (that thus generate cross-ISP traffic), the percentage of videos served by P2PTube enabled boxes (that thus avoid cross-ISP traffic), as well as the percentage of videos that are cached by some P2PTubers that are however busy serving other peers (blocked). Two solid lines represent online averages for cache hit and blocking rates.

First, notice that about 46% of videos could be served by P2PTubers, dividing by two the amount of inter-ISP bandwidth due to YouTube entertainment (about 10% higher than in [10] whose results are affected by peer churn and by the absence of video advertisement). Second, the P2PTube blocking probability is low, about 3% on average, and *decreasing*: this is due to the fact that as popular videos spread over multiple peers, a larger number of copies are available so that after an initial transient blocking fades away.

Next, we report in Tab. I a sensitivity analysis of the cache hit rate for varying cache size (10MB, 100MB, 1GB) and upload rates (100Kbps, 500Kbps, 5Mbps). Obviously, cache hit improves with cache size and upload rates (though improvement tends to saturate, as even with infinite cache size, the first requests would anyway be a miss).

More interestingly, a non marginal caching benefit (20-25% in our trace) happens even in the extreme case where

TABLE I. SENSITIVITY ANALYSIS OF P2PTUBE CACHE HIT PROBABILITY

Cache	Upload bandwidth		
	100 Kbps	500 Kbps	5 Mbps
10 MB	0.22	0.25	0.25
100 MB	0.36	0.46	0.46
1 GB	0.42	0.53	0.54

P2PTube only stores about a single video (e.g., 10MB RAM cache for diskless systems). Here, caching benefits is mainly due to the initial advertisement videos shown *before* the user requested videos: as advertisement are often the same on a given day/location irrespectively of the user-selected video, this implies high caching efficiency even with tiny caches.

We next consider that, due to user activity, only a small portion of the upload capacity is available to P2PTube (e.g., 100Kbps). This corresponds to a stress-test of the system where upload rates are lower than the stream rate: while in this case P2PTube would introduce an initial buffering delay (and thus unacceptable QoE for the user) it is still interesting to observe that the impact on the cache hit probability is modest. This cache hit reduction is due to the fact that at lower rate, video upload takes longer, so that odds that subsequent requests for popular videos arrive before the previous upload has finished increases. Yet, as we see the limited cache hit reduction suggests this not to be a problem, as download durations are not as long as in file-sharing, nor are arrivals as correlated as in P2P-TV and live-streaming.

IV. AFTERMATH

P2PTube is a simple yet effective peer-assisted system for caching of YouTube that, leveraging set-top-box ubiquity and persistence, can cut the cross-ISP traffic by half (in our trace). As in our current design the upload is entirely assured by a single peer, future work is necessary to verify these caching benefits to hold in presence of user background traffic – or if parallel download and chunking are necessary to guarantee streaming rate in presence of real user traffic.

REFERENCES

- [1] H. e. a. Chang, “Live streaming performance of the zattoo network,” in *ACM IMC*, 2009.
- [2] <http://youtube-global.blogspot.com/2011/05/thanks-youtube-community-for-two-big.html>.
- [3] A. a. a. Finamore, “Youtube everywhere: Impact of device and infrastructure synergies on user experience,” in *ACM IMC*, 2011.
- [4] Youtube sucks on french isp free, and french regulators want to know why. <http://gigaom.com/2013/01/02/youtube-sucks-on-french-isp-free-french-regulators-want-to-know-why/>.
- [5] An ad-block shock. france v google: Xavier niel is playing rough with the internet giant. <http://www.economist.com/news/business/21569414-xavier-niel-playing-rough-internet-giant-france-v-google>.
- [6] W. e. a. Jiang, “Orchestrating massively distributed cdns,” in *ACM CoNEXT*, 2012.
- [7] P. e. a. Gill, “Youtube traffic characterization: a view from the edge,” in *ACM IMC*, 2007.
- [8] X. Yang and G. de Veciana, “Service capacity of peer to peer networks,” in *IEEE INFOCOM*, 2004.
- [9] P. Rodriguez and E. W. Biersack, “Dynamic parallel access to replicated content in the internet,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, 2002.
- [10] M. e. a. Zink, “Characteristics of youtube network traffic at a campus network - measurements, models, and implications,” *Comput. Netw.*, vol. 53, no. 4, pp. 501–514, Mar. 2009.