Impact of Caching on HTTP Adaptive Streaming Decisions: towards an Optimal

Vitalii Poliakov, Lucile Sassatelli
Université Nice Sophia Antipolis
Email: {last}@i3s.unice.fr

Damien Saucez
Inria
Email: {first.last}@inria.fr

***Abstract*—The interplay between caching and HTTP Adaptive Streaming (HAS) is known to be intricate, and possibly detrimental to QoE. In this paper, we make the case for caching-aware rate decision algorithms at the client side which do not require any collaboration with cache or server. To this goal, we introduce the optimization model which allows to compute the optimal rate decisions in the presence of cache, and compare to this optimal, the current main representatives of HAS (RBA and BBA). This allows us to assess how far from the optimal these versions are, and on which to build a caching-aware rate decision algorithm.**

***Keywords*-HTTP adaptive streaming, video caching, QoE.**

I. HTTP ADAPTIVE STREAMING (HAS)

With HTTP adaptive streaming (HAS), the video player decides which rate to request for each video chunk. Different video rate decision algorithms exist and consider different decision criteria. We can isolate two main families: *rate-based* (RBA) and *buffer-based* (BBA) algorithms, which both try to maximize the video bit rate by fitting it to the estimated network bandwidth for RBA, and to the buffer occupancy for BBAs (to avoid stalls as much as possible) [1]. As in HAS each chunk is a content retrieved with HTTP, it means that the various chunks can be cached anywhere on an HTTP cache.

As a result, a well-known oscillation problem arises when HAS interacts with in-network caches: some chunks may be cached and hence stored closer to the video player while others are not, making the video rate selection algorithms erroneously decide to increase video rate playback [2]. A number of possible solutions have been recently presented in the literature. A number of them rely on traffic shaping to indirectly control the decision algorithm, e.g., [2], [3], requiring modifications at the server side and possibly at the cache side. As well, Cache'n Dash [4] modifies the cache so that it predicts client rate requests and pre-fetches the necessary chunks. Unlike [4], Gearbox [5] strives to avoid any server or cache modification, and instead devises a client algorithm basing its decisions on the playout buffer as in [1], but refines it by modulating the aggressiveness on occupancy intervals to account for a possible cache's presence. Despite Information Centric Networks (ICN) and client-CDN collaboration are assumed in the above works (except [5], they are still in the experimental stage.

The goal of our work is to devise efficient rate decision algorithms which account for caching and content popularity, by building on a principled optimization approach to the different QoE metrics, while not relying on ICN or additional client-cache-server signaling. In this paper, we present the first step towards this goal, which consists in introducing the optimization model which allows to compute (in an oracle-way) the optimal rate decisions in the presence of cache, and compare to this optimal, the current main representatives of RBA and BBA. This allows us to assess how far from the optimal these existing simple versions are, and on which to build a cache-aware rate decision algorithm. The conclusion explains how we intend to proceed from these results.

II. EXPERIMENTAL ENVIRONMENT

The testbed is a chain topology made of three virtual GNU/Linux machines: an Apache HTTP server, an instrumented video player based on VLC v.3.0.0 and a Squid proxy acting as a transparent cache. The *tc-netem* Linux tool from the *traffic control* suite is used to emulate a cache-server link of capacity $C_s = 2$Mbps while the link between the cache-client link is non-constrained to $C_c = 1$Gbps. For all the experiments we stream the *Big Buck Bunny* video[1] that we encoded in HLS with the bitrate of the maximum quality representation higher than C_s. This video is made of $K = 300$ chunks of 2s-duration each. Each point of the next figures is generated with 15 samples, and the 95%-confidence intervals are shown.

III. QOE METRICS

Quality of Experience (QoE) metrics are meant to represent users' perception of the video playback. Despite the intrinsic subjectivity of such experience, the multimedia community has agreed on three most important metrics for video, as defined in [6]: rebuffering ratio (representing the impact of playback stalls), video quality (a log function of the video bit rate) and quality instability. As aforementioned, these metrics are taken into account differently by different rate decision algorithms. Let us mention we consider BBA2 and BBA3 in the next section (they are defined in [1]). BBA2 is more aggressive in ramping-up the video rate while

[1]Available at https://peach.blender.org

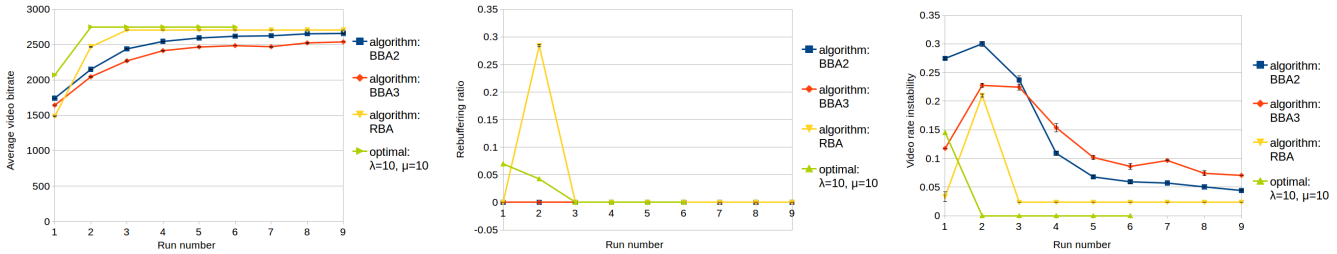

Figure 1: Comparison of RBA and BBAs to the optimal policy for all three QoE metrics.

BBA3 moderates the aggressiveness by accounting for the rate instability too. To unify these approaches, Yin et al. in [6] approached in a principled way the rate-decision problem as an optimal control problem. In particular, they defined QoE_k as the weighted sum of the above three QoE metrics above, for chunk $k = 1, \ldots, K$. The weights (see [6]) are 1, $-\lambda$ and $-\mu$ for the video quality, quality instability and rebuffering ratio, respectively. The objective function considered is then $\sum_{k=1}^{K} QoE_k$. Let R denote the number of clients successively requesting the same video (r is the index of the video request/runs). We extend this formulation to the caching problem, with KR rate decision variables, and consider the problem $\max \sum_1^R \sum_1^K QoE_{r,k}$, which corresponds to maximizing the sum of QoE over all the R clients.

IV. Experimental results

Comparing run 1 when no cached chunks are available with next runs, Fig. 1 shows as expected that caching improves video bit rate. By being more aggressive, RBA converges faster to the maximum rate, while both BBA1 and BBA2 lag behind. This faster convergence of RBA comes however at the expense of stalls during playback, in particular in the second run. By construction, BBAs strive to avoid stalls primarily. The video bit rate oscillations due to caching are seen in the high instability over the first runs. In agreement to its very design, BBA2 has a lower instability than BBA2 when the downloading bandwidth varies over consecutive chunks (i.e., in the first runs), at the expense of a slightly lower bit rate.

The optimal QoE values are obtained with the above model where both instability and rebuffering ratio are accounted for with weight $\lambda = \mu = 10$. We observe that these parameters allow to obtain a video rate consistently better over all runs (consecutive clients). Interestingly, the instability and rebufferings are moved to the first and second clients: this explains as the optimization is performed knowing the number of runs/clients. So in order to maximize the average QoE over all clients, it is better to penalize a bit the first clients in anticipation of the next ones. Doing so allows to fetch higher quality chunks in the cache, in benefit of the next clients. Despite the apparent unfairness of such decision (note that the client fairness is not part of the optimization), it actually benefits the majority, thereby unveiling that the instrumentation of rate decision can be a straight-forward path to control cache state. Note that the chosen example is rather extreme, as stalls are undergone by the first two clients, which can compel them to abandon viewing. This can be easily avoided by changing the weigths of rebufferings in the optimization.

V. Conclusion

By carrying out an experimental study to compare the impact of caching on QoE when RBA or BBAs are employed at the client side, and comparing them to the optimal policy, we intend to make the case for cache-aware rate decision algorithms at the client side. In particular, the number and frequency of consecutive clients (runs above) directly connects to the content popularity. By adding only popularity information to the manifest file, without any other additional signaling (from cache or server, without shaping), we shall devise adaptive client rate decision. Their distance from optimal will assess how acute is the need for further entity collaboration. We plan on building lookup tables from binning the results of the above caching-aware optimization model.

References

[1] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *ACM SIGCOMM Comp. Comm. Review*, vol. 44, no. 4, pp. 187–198, 2015.

[2] D. H. Lee, C. Dovrolis, and A. C. Begen, "Caching in http adaptive streaming: Friend or foe?" in *ACM NOSSDAV*, 2014.

[3] C. Kreuzberger, B. Rainer, and H. Hellwagner, "Modelling the impact of caching and popularity on concurrent adaptive multimedia streams in information-centric networks," in *IEEE ICME Workshops*, June 2015.

[4] P. Juluri and D. Medhi, "Cache'n dash: Efficient caching for dash," in *ACM SIGCOMM*, 2015, pp. 599–600.

[5] H. Yunfeng, "Cache-friendly Rate Adaptation for DASH," May 2015.

[6] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *ACM SIGCOMM*, 2015, pp. 325–338.