IMPROVING USER EXPERIENCE WHEN HTTP ADAPTIVE STREAMING CLIENTS COMPETE FOR BANDWIDTH

Ch. Taibi, F. Le Bolzer, R. Houdaille

Technicolor, France

ABSTRACT

The last decade has witnessed a tremendous increase of high-definition video consumption over the Internet and the emergence of the HTTP adaptive streaming technique, designed to cope with highly varying delivery conditions. Despite all its inherent advantages, HAS suffers from weaknesses when competing for bandwidth. In this paper, we focus on improving the HAS behavior for several cases of bandwidth competition. We propose in the first part a dynamic traffic shaping method operating in two phases (transient and steady) to improve convergence speed when HAS competes with a greedy TCP flow. However, when competition takes place between several HAS clients, this solution sub-optimally uses the bandwidth. Therefore, the second part investigates HAS client collaboration, defining information to be exchanged between clients and a common set of rules to determine the best representation to be requested.

INTRODUCTION

The increasing number of connected displays is driving the current explosion of Internet video traffic. In this context, streaming over HTTP has become the dominant approach for delivering multimedia content over the Internet. Several versions of HTTP adaptive streaming have been proposed by various stakeholders and pushed for the provision of “over the top” audiovisual delivery in the Internet. In addition, standardization efforts have been made. Since 2012, the MPEG Dynamic Adaptive Streaming over HTTP [1] (MPEG-DASH) standard is available. Such adaptive protocols will be referred to as “HTTP Adaptive streaming” or HAS throughout this paper. While these techniques have been designed to cope with varying delivery conditions, under some specific challenging circumstances, HAS client implementations suffer from severe performance issues, namely instability, unfairness and bandwidth underutilization.

The first part of the work is dedicated to HAS competition with a greedy TCP flow. We propose a central dynamic traffic shaping method to mitigate the convergence issue and to optimize HAS client convergence. The second part addresses HAS competition among several clients through implementing a distributed algorithm. We define information to be exchanged between clients and a common set of rules that enables each client to determine the best representation to be requested. The benefits of this solution will be evaluated thanks to an implementation based on MPEG-DASH standard [1]. Finally, the paper presents the standardization effort made to include this technique into MPEG-DASH.

RELATED WORK

The main idea of HAS techniques is to ensure a continuous playback by choosing representations whose bit-rate fit the average available bandwidth. However this available bandwidth is generally only estimated by the client from the received rate of the content. When several clients receive data at the same time, TCP allows to provide a fair share of the bandwidth to the clients. But since HAS content is segmented, each client has periods of 'full speed' download interleaved with periods of no activity. When several clients operate on the same local network, the way these (variable) periods overlap influences the perceived TCP throughput. The clients may see instable measured bandwidth which leads to potential user experience degradation such as unstable choice of representation or unbalanced share of bandwidth between competing clients.

This issue, known as the 'downward quality spiral phenomenon', has been characterized in [2] and explained in [3] and [4]. To mitigate it, two strategies may be used: improve the HAS implementation or manage the content delivery from the network infrastructure.

Regarding the HAS implementation first, [4] and [8] proposed some improvements at the client side. The main idea consists in reducing the client conservatism. The client becoming greedier has a better chance of grabbing bandwidth. Nevertheless greedy clients overreact to the bandwidth variations [10]. A second idea, investigated in [8], proposes to modify the segment request scheduling to randomize the ON-OFF periods and then break the downward spiral. Implementation improvements have also been proposed on the server side. S. Akhshabi and al. 9 try to tackle the instability issue modifying the HAS server to detect oscillating clients and to shape the rate of their requested segments to reduce as much as possible the OFF period duration causing the instability. Nevertheless to work correctly, all the segments should be handled by the same server which is quite restrictive with regard to CDN infrastructures.

Regarding the network infrastructure next, authors in [11] have proposed to manage the content delivery of concurrent HAS sessions in a home network by determining the best representation based on a Quality of Experience (QoE) criteria. They inform network elements of the bandwidth management decisions and the HAS clients of the targeted representation. This approach is interesting since it introduces a way to determine the targeted bitrate of a streaming session and a way to communicate it to the respective client. However, the proposed implementation may be costly and lacks of details on how bandwidth management rules are calculated. Another paper [12] has also focused on the QoE, given the control to the network operator. This solution consists in rewriting HAS client requests, keeping the client unaware of the modifications, which is not desirable: HAS client may need to know what it receives to work correctly. In [2] and [11], authors have proposed to implement bandwidth management in the home gateway using a traffic shaping method. The method defines a target representation for each HAS stream and then constrains the clients to stay in these bandwidth limits. This solution has two advantages: it handles any HAS implementation and it tackles both the stability and the fairness issues. However, it does not guarantee the targeted representation nor does it manage the convergence speed.

IMPROVE CONVERGENCE SPEED AND STABILITY OF HAS VS. GREEDY FLOWS

The traffic shaping method proposed in [2] controls the competition between several HAS clients. We propose to apply it to a competition scenario between a HAS stream and a pure download and improve it to take into account the convergence time. The use of traffic

shaping aims at providing a deterministic streaming experience with regard to **the targeted video representation, the convergence speed and the stability**.

Experimental Setup

Our test bench is depicted in Figure 1. It reproduces real home network environment with a bottleneck on its WAN interface. All data flows are streamed from the Internet. For statistical purposes, at least 50 experiments (each lasting 4mn) are repeated on each considered scenario and the results are averaged.

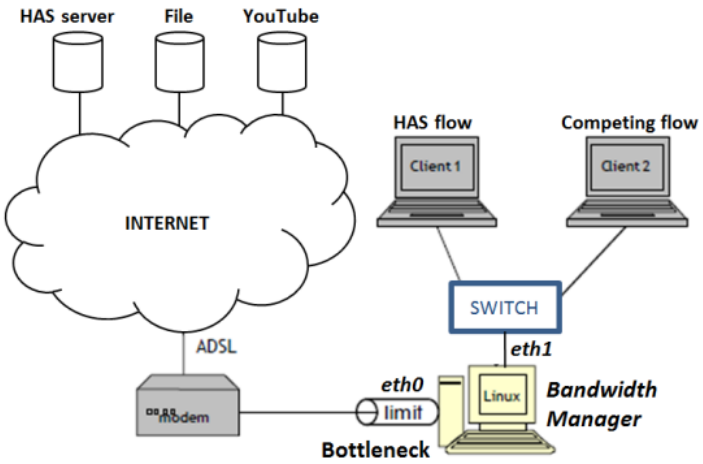

Figure 1 - Test Bench description

HAS convergence study

Before defining the new shaping scheme, we focus on the HAS startup to study its convergence to a given target rate. The HAS player (client1) was characterized while playing alone on different bottleneck conditions, translated into a conservatism margin ratio (***Cons_margin***), defined by (Eq. 1).

Cons_margin = (BW – target) / BW (Eq. 1)

BW is the available bandwidth and ***target*** is the targeted video representation. Here it is fixed to the maximum available representation. Two HAS technologies have been tested: Microsoft Smooth Streaming (Smooth) [5] and Apple HTTP Live Streaming (HLS) [6]. Note that Smooth and HLS players apply a conservatism value of 20% and 40% respectively [8] when selecting the segment representation with regard to the measured bandwidth.

The convergence results are presented in Table 1 for both technologies. The convergence time is defined as the average time required by the player to hit the target over the total number of successful experiments. A hit is validated if the target was reached AND maintained during the rest of the experiment.

Bottleneck (kbps)	Cons_margin (%)	Hit ratio (%)	Convergence time (avg. in s)
Smooth (target= 2436kbps)			
2700	10	88	48
3050	20	100	23
4050	**40**	**100**	**1**
5700	**60**	**100**	**<1**
HLS (target= 1580kbps)			
3150	50	60	127
3550	**55**	**100**	**10**
5700	**70**	**100**	**3**

Table 1 - Smooth and HLS convergence dynamics.

The Smooth player manages to reach the target with a conservatism margin ratio equal to 20%, which is its conservatism value, but the convergence is slow (23s). Applying a higher margin value greatly improves the convergence speed (1s with cons ≥ 40%). On the other hand, the HLS player needs a conservatism margin ratio equal to 55% to ensure the convergence, which is higher than its conservatism (40%). Similarly to Smooth, higher margin values further accelerate the HLS convergence.

Introducing dynamic shaping

In [2], the proposed traffic shaping method manages the competition between 2 HAS streams. The shaping consists in defining two service classes: one for each HAS stream, each being assigned a set of QoS parameters. The bandwidth manager is implemented using the Linux tc tool with the HTB (Hierarchical Token Bucket) queuing discipline [13]. We apply it to our competition scenario with one service class for each HAS stream under control and another one for other traffics. The HTP *rate* parameters are then defined by (Eq. 2) and (Eq. 3), where ***target*** is the desired representation level and ***cons*** is equal to the HAS player conservatism value: 20% and 40% for Smooth and HLS, respectively [7].

$$rate_{HAS} = target \times 100 / (100-cons) \quad \textbf{(Eq. 2)}$$

$$rate_{other} = bottleneck - \Sigma rate_{HAS} \quad \textbf{(Eq. 3)}$$

As demonstrated before, such a shaping doesn't ensure a quick HAS convergence (Table 1). To improve the convergence speed, the ***cons*** value of (Eq.2) should be higher than the player conservatism. On another hand the margins should be as small as possible to optimize bandwidth usage. We solved this dilemma by introducing a dynamic scheme implementing two shaping phases. The first phase, called "***transient shaping"*** uses a high margin and is applied for a short time, to reach the target. The second phase, called "***steady shaping"***, uses a margin equal to the client conservatism and is applied as long as the HAS player requests the targeted representation. Both shaping phases use (Eq. 2) and (Eq. 3) but with 2 different margin values: $cons_{trans}$ and $cons_{steady}$ respectively. Assuming the HAS player conservatism is known, the transient margin, $cons_{trans}$, should be chosen higher than that value while the steady margin, $cons_{steady}$, will be equal to it. **For the Smooth technology, we recommend to choose $cons_{trans}$ between 40% and 60% and for HLS, we recommend to choose it between 55% and 70%.**

To complete the dynamic scheme we defined a third parameter: the transient shaping duration (D_{trans}). Additional experiments were performed to determine its optimum value. For each HAS technology, two values of $cons_{trans}$ have been tested. They were chosen to provide a fast convergence. The results obtained under different D_{trans} values are presented in Table 2**Error! Reference source not found.**. The D_{trans} values have been defined to range from a value just above the expected convergence time (defined from Table 1), to higher values.

We observed that to guarantee convergence, the transient shaping phase should be maintained beyond the convergence time. Indeed, if the steady shaping occurs too early, the player tends to cut down the segment representation after the transition. If we assume that the buffer is entirely filled during the transient shaping, the buffering time can be estimated by (Eq.4).

$$Buffering\ Time = L_{Buf} \times (100-cons_{trans}) / cons_{trans} \quad \textbf{(Eq. 4)}$$

L_{Buf} is the HAS client playback buffer length expressed in seconds. Its value was chosen equal to 30s for both HAS technologies as found in the literature [5] and confirmed by our experiments. The estimated buffering time values are presented in Table 2. It is worth noting the correlation between the HAS client buffering time and the transient duration, required to guarantee a good convergence. Both decrease when the transient margin (***$cons_{trans}$***) increases. **To guarantee the best convergence, we recommend maintaining the transient shaping as long as the HAS client playback buffer is not filled**.

D_{trans} (s)		5	15	30	45	Expected convergence time (s)	Estimated Buffering Time (s)
Smooth (target=2436kbps, $cons_{steady}$ = 20%)							
$Cons_{trans}$ = 40% $Rate_{HAS}$ = 4050kbps	Hit ratio (%)	67	38	100	100	1	45
	Convergence time (s)	36	30	1	1		
$Cons_{trans}$ = 57%$Rate_{HAS}$ = 5700kbps	Hit ratio (%)	22	100	100	100	<1	22
	Convergence time (s)	51	17	<1	<1		
HLS (target= 1580kbps, $cons_{steady}$ = 40%)							
$Cons_{trans}$ = 55% $Rate_{HAS}$ = 3550kbps	Hit ratio (%)	57	82	100	N.A.	10	24
	Convergence time (s)	2	5	8	N.A.		
$Cons_{trans}$ = 72% $Rate_{HAS}$ = 5700kbps	Hit ratio (%)	100	100	100	N.A.	3	12
	Convergence time (s)	2	2	2	N.A.		

Table 2 - Smooth and HLS convergence with dynamic shaping.

Experimental results

The dynamic shaping scheme has been tested on competition scenarios between a HAS stream and a pure download (DL). Two startup sequences have been considered, each one highlights specific aspects of the HAS streaming behavior:

1. **DL first**: the download starts alone, and the HAS flow starts after 30 seconds. This first scenario illustrates the HAS client ability to start in a competition context and to win back bandwidth from an established flow.

2. **together**: both flows start together, allowing the comparison of their respective greed for bandwidth.

The results are presented for both HAS technologies (smooth and HLS) in Table 3 and in Table 4 respectively. They compare the convergence performances obtained in the three

shaping conditions: no shaping, static shaping and dynamic shaping. The performances are defined with regard to the convergence to the maximum available representation.

Shaping mode	**NONE**	**STATIC** cons=***cons*$_{steady}$**=20%**		**DYNAMIC** cons$_{trans}$**=40%**, D$_{trans}$**=45s**	
scenario	Hit ratio	Hit ratio	Conv. Time	Hit ratio	Conv. Time
DL first	0%	100%	25s	100%	3s
together	0%	100%	25s	100%	7s

Bottleneck = 5700kbps, target = 2436kbps, cons$_{steady}$=20%

Table 3 - Smooth competing with download.

Shaping mode	**NONE**	**STATIC** cons=***cons*$_{steady}$**=40%**		**DYNAMIC** cons$_{trans}$**=55%**, D$_{trans}$**=30s**	
scenario	Hit Ratio	Hit Ratio	Conv. Time	Hit Ratio	Conv. Time
DL first	6%	2%	N.A.	100%	7s
together	10%	4%	N.A.	100%	7s

Bottleneck = 5700kbps, target = 1580kbps, cons$_{steady}$=40%

Table 4 - HLS competing with download.

Applying the dynamic shaping according to the schemes defined previously allows the HAS client to converge quickly and surely to the targeted representation. The convergence time is much shorter than with the static shaping while the bandwidth allocation remains very similar considering the content duration.

These results validate the dynamic shaping rule and demonstrate its capacity to manage the HAS client convergence from 3 shaping parameters: ***cons$_{trans}$***, ***cons$_{steady}$***, and ***D$_{trans}$***. Nevertheless, this approach leaves open issues: a) how the target can be determined by the gateway; b) it might be inappropriate to define many traffic shaping rules when several HAS clients operate on the same home network; c) when the competition occurs between HAS clients only, giving margin to one may have a negative effect on the other(s). That's why we investigate another way of improvement that allows defining the targeted representation and that prevents the gateway to set-up many shaping rules.

COOPERATION BETWEEN HAS CLIENTS

When starting a streaming session, a HAS client has to determine the optimal representation, but generally, it has little information on the throughput it can expect. Algorithms based on observation of the actual reception rate allow converging but require some delay to do so. On one hand if the initial representation is chosen at a low bit-rate, the visual quality will be low in the first seconds of a video. On the other hand if it is chosen too high, either buffering may delay the start of the video, either download may be too slow causing a video freeze. Hence the client would benefit from assistance in selecting the ideal representation as soon as possible. Different strategies and means may be used to deliver hinting information to the clients. One possibility is to inform the client through the manifest file, but it requires a modification "on-the-fly" of the manifest which may be costly and which is even not possible with an https connection.

Another possibility, developed below, is to inform clients of the local network characteristics (available bandwidth, list of concurrent clients and their session characteristics) and let them decide of the best representation to choose.

Cooperation principle

The basic idea is that a group of HAS clients in the same home network exchange parameters describing their respective sessions. While still using its usual algorithm for the desired representation, each client uses the knowledge of other sessions to prevent himself from taking too much bandwidth, so that a fair sharing of the access link bandwidth is cooperatively obtained.

Ideally the total amount of bandwidth to share is signalled from the home gateway. Indeed the gateway has a good view on the actual properties of the access link, and also may allocate bandwidth to different types of traffic and thus decides how much is useable by HAS sessions. Otherwise, HAS clients have to estimate the bandwidth by other means.

All clients should conform to a common set of rules providing consistent results. Depending on the desired philosophy, different algorithms are possible.

Needed cooperative parameters

To cooperate, HAS clients have to exchange several parameters, described hereafter.

sessionDescription: this parameter groups HAS session information allowing clients to implement a cooperative behavior when sharing bandwidth on the home network. It includes:

- A unique session identifier
- The list of needed bandwidth for all representations that are relevant for the given client
- The session priority which is used to influence the bandwidth distribution according to users' preferences (e.g. one can give a higher priority to the main TV set over all other devices in the home to favour it for a larger share of the bandwidth).
- Session status (playing, paused, ended)

sharedBandwidth: the bandwidth part of the home access link which is shared among several HAS clients. This parameter allows the clients to know the collectively shared bandwidth they have to consider when making a cooperative decision.

If several algorithms are possible, they have to be identified and an additional parameter (**preferredAlgorithm**) can be used to do so. It is also possible to use a predefined algorithm to determine the collaborative algorithm that must be used by all the collaborative clients. As an example, the selected algorithm will be the one given by the oldest session in the highest priority list.

Collaborative algorithm

Several algorithms are possible. As an example, the principles of the so called « Premium privileged » algorithm are:

- Sessions of highest priority take a fair share of the sharedBandwidth
- The rest of the sharedBandwidth (if any) can be used by the group of lower priority.

MPEG-DASH standardization

In order for the cooperation to be effective between devices from several vendors, the required parameters and the algorithm have to be shared by all involved HAS clients of the local network. MPEG has already released a standard, DASH - part 1 [1] to define how adaptive streaming content have to be delivered. More recently the SAND (Server And Network Assisted DASH) core experiment was launched to standardize means of improving delivery through assistance of the client.

ON-GOING AND FUTURE WORK

The cooperative solution has been implemented for MPEG-DASH content, running on Android platforms. We have planned to make field tests during the second semester of 2015 to evaluate the improvements provided by this solution and collect user feedbacks.

CONCLUSION

In this article we have focused on the known issues of HAS clients competing for bandwidth over the home broadband access and introduced two new tools to mitigate these issues. One extends the idea of traffic shaping in the home gateway with a dynamics improving convergence for starting HAS sessions. The other is based on explicit cooperation between HAS clients within a home, actually removing the competition.

REFERENCES

1. ISO/IEC IS 23009-1, “Information technology- Dynamic adaptive streaming over HTTP (DASH)- Part I: Media presentation description and segment formats”, Apr. 2012.
2. R. Houdaille, S. Gouache, “shaping HTTP adaptive streams for a better user experience”, *ACM MMsys,* 2012.
3. S. Akhshabi, C. Dovrolis and A. C. Begen, “What Happens When Adaptive Streaming Players Compete for Bandwidth?”, *ACM NOSSDAV*, 2012.
4. T.Y. Huang, N. Handigol, B. Heller, N. McKeown and R. Johari, “Confused, Timid and unstable: Picking a video streaming rate is hard”, *ACM IMC*, 2012.
5. Microsoft publications, “Smooth Streaming Protocol specifications”, [MS-SSTR] - v20131025, October 2013.
6. R. Pantos, “HTTP Live Streaming”, IETF Internet-Draft Version 10, October 2012.
7. S. Narayanaswamy, A. C. Begen and C. Dovrolis, “An experimental evaluation of rate-adaptive video players over HTTP”, *SIGNAL PROCESS-IMAGE*, 27:271-287, 2012.
8. J. Jiang, V. Sekar and H. Zhang, “Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE”, *ACM CoNEXT*, 2012.
9. S. Akhshabi, L. Anantakrishnan, C. Dovrolis and A. Begen, “Server-based Traffic Shaping for Stabilizing Oscillating Adaptive Streaming Players”, *ACM NOSSDAV*, 2013.
10. R. R. Luciano “*A Dynamic Adaptive http Streaming Video Service for Google Android*”, Thesis Royal Institute of Technology Stockholm, 2011.
11. P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, N. Race “Towards Network-wide QoE Fairness Using OpenFlow-assisted Adaptive Video Streaming”, *ACM SIGCOMM* 2013 Workshop on Human-Centric Future Multimedia Networking.
12. El Essaili, D. Schroeder, D. Staehle, M. Shehada, W. Kellerer and E; Steinbach “Quality-of-Experience driven Adaptive http Media Delivery” IEEE ICC 2013.
13. [HTB] Martin Devera, “HTB Homepage, http://luxik.cdi.cz/~devik/qos/htb/