



HAL
open science

MS-Stream: A multiple-source adaptive streaming solution enhancing consumer's perceived quality

Joachim Bruneau-Queyreix, Mathias Lacaud, Daniel Negru, Jordi Mongay Batalla, Eugen Borcoci

► To cite this version:

Joachim Bruneau-Queyreix, Mathias Lacaud, Daniel Negru, Jordi Mongay Batalla, Eugen Borcoci. MS-Stream: A multiple-source adaptive streaming solution enhancing consumer's perceived quality. IEEE Consumer Communications and Networking Conference - CCNC, 2017, Las vegas, United States. hal-01504005

HAL Id: hal-01504005

<https://hal.science/hal-01504005>

Submitted on 7 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MS-Stream: A multiple-source adaptive streaming solution enhancing consumer’s perceived quality

Joachim Bruneau-Queyreix^{*†}, Mathias Lacaud[†], Daniel Negru^{*}, Jordi Mongay Batalla[‡], Eugen Borcoci[§]

^{*} Univ. Bordeaux, LaBRI, UMR 5800
F-33400 Talence, France
{jbruneau,negru}@labri.fr

[†]Viotech Communications
Versailles, France
{jbruneau,mlacaud}@viotech.net

[‡]Warsaw University
of Technology
jordim@interfree.it

[§]University Politehnica
of Bucarest Romania
eugen.borcoci@elcom.pub.ro

Abstract—Adaptive bitrate streaming protocols, such as DASH, have seen extensive interests for their adaptation capabilities to increase consumers’ Quality of Experience (QoE) over the Internet, and have become de-facto standards in web video delivery. Compared with traditional single-server approaches, multiple-server streaming offers the opportunity to exploit expanded bandwidth, link diversity, and reliability. In this paper, we expose our solution for multiple-server support to dynamic adaptive streaming applications: Multiple-Source Streaming (MS-Stream). Thanks to its codec agnosticism and DASH-compliance our contribution is a pragmatic and evolving solution for QoE enhancement that can be applied to many streaming architectures (CDNs, Clouds) and is particularly suited for distributed environments such as P2P or Set-Top-Box overlays. In addition, splitting content into multiple independent sub-streams provides the opportunity to achieve easy-to-design bitrate adaptation and server-switching mechanisms. We empirically validate our approach using an extensive collection of network profiles provided by the DASH Industry Forum. Our solution is compared with the full potential of DASH with several servers over several QoE criteria. Results show the QoE gain of using MS-Stream against DASH; an online demonstration is made available.

I. INTRODUCTION

For several years, video traffic over the Internet has had a tremendous increase, becoming the most popular service on the web [1]. Most of the existing video streaming solutions use Over-The-Top (OTT) techniques based on HTTP-based adaptive streaming protocols with one server only (such as MPEG-DASH [2] or HLS [3]) where clients can decide to adapt the content bitrate delivered by a server to estimated network resource status. The flexibility brought by such adaptive streaming protocols permits to avoid video freezing events and increases the consumers’ perceived Quality of Experience (QoE). Using HTTP for video streaming is a strong asset in easing the adoption of innovative streaming solutions. Indeed, it significantly simplifies the traversal of firewalls and network address translation. Furthermore, the deployment cost of HTTP-based solution is relatively low since it relies on a standard technology and, therefore, enables a smooth integration into existing infrastructures such as CDNs, Clouds and other distributed systems.

In this paper, we propose an evolution of existing single-server approaches by considering the simultaneous use of multiple sources for one video. Each content source can stream an independent decodable and playable sub-stream (created from the already encoded content) that provides sub-optimal video quality. Sub-streams (referred as *descriptions*

[4]) can be combined at client side and the original video quality can be reconstructed in order to provide consumers with an increased perceived video quality. Targeting quality enhancement through the use of multiple servers and path diversity, we propose the intra-description switching algorithm integrating bitrate adaptation and server-switching mechanism.

Bringing multiple-source support to existing HTTP-based adaptive streaming solutions not only represents the immediate opportunity to take advantage of available bandwidth on different network paths and servers, but also to develop new types of multimedia streaming applications. For instance, our Multiple-Source Streaming (MS-Stream) solution permits almost all existing devices that have network connectivity to become part of streaming sessions and to bring enhancement in the final perceived video quality of consumers (in accordance with device network connectivity: -adsl2+, optical fiber, vdsl, etc.-). Therefore, streaming providers would no longer have to fully rely on high-cost over-provisioned server infrastructures (in CPU, RAM, bandwidth), but could also use other existing devices acting as content sources that would eventually provide better perceived quality to consumers.

MS-Stream is a pragmatic solution achieving multiple-server support in OTT distributed streaming systems. Thanks to its codec agnosticism and backward DASH-compliance, this proposal represents an evolving solution that can be applied to all streaming architecture models (P2P, CDNs, Clouds), achieving lightweight easy-to-design multiple-server content delivery. The key components of the proposal lie in innovations brought at the coding/decoding scheme and within the streaming protocol, along with its associated algorithm. We evaluate the MS-Stream solution over different network profiles with its real-world implementation built on top of the open-source DASH software provided by the DASH-Industry Forum [5]. The conducted evaluation exposes the QoE¹ gain of using MS-Stream over the de-facto standards currently used for content delivery over the Internet. It ought to be noted that the MS-Stream solution comes with a network bandwidth consumption overhead, also evaluated and discussed.

The rest of the paper is organized as follows. Section II provides some background and related works on content

¹In this paper QoE gain is referred as providing the highest possible content bitrate (in a specific codec format) while minimizing video freezing events and lowering quality fluctuation amplitudes, which are the most important QoE metrics of adaptive streaming protocols.

delivery in multiple-source environments. Section III describes the MS-Stream proposal at the codec and streaming protocol levels. Section IV presents empirical results obtained on QoE evaluation. We conclude in section V. An online prototype is implemented and available at [6].

II. RELATED WORK

Adaptive bitrate streaming protocols, such as DASH and HLS, have seen extensive interest from the industry and the research community, mainly due to their capabilities to render smooth video playback to the consumers. Thus enhancing QoE. The author in [7] introduced the DASH framework of Netflix, which has been the largest DASH stream provider in the world and the work [8] outlined that Netflix always binds a user to one server, regardless of the available throughput between the user and the server. The study performed in [8] also indicates that QoE could greatly benefit from simultaneously using multiple servers along with DASH-like protocols.

Bitrate adaptation mechanisms represent the most important research challenges related to dynamic adaptive streaming applications. A tremendous amount of papers have been published on this topic. The work in [9] compared the bitrate adaptation of three commercial clients: Microsoft Smooth Streaming [10], Adobe OSMF [11], and Netflix client [7]. Algorithms were suggested to increase QoE in single-server HTTP adaptive streaming. For example, bandwidth-based bitrate adaptation [12][13][14] and buffer-based bitrate adaptation [15]. However, our approach is based on simultaneous segment retrieval from several source servers, requiring new methods for bitrate adaptation, where classic single-server algorithms cannot be applied. A preliminary bitrate adaptation approach to be extended to multiple sources is proposed in [16]. Originally not suited for a multiple servers purpose, the proposed bitrate adaptation decisions are asynchronously taken, imposing segments to be retrieved one after another, without considering completion time. Therefore, request completion is most of the time out-of date due to channel heterogeneity in multiple- server environments. In contrast with this approach, our proposed solution considers requesting several servers simultaneously and canceling segment requests which cannot meet their assigned deadline. In addition, MS-Stream relies on the delivery of other sub-streams to provide smooth content playback, yet in sub-optimal quality.

The contribution in [17] presents Presto, a streaming protocol designed to use several servers simultaneously in order to improve QoE by providing better fairness, efficiency and stability at the service provider's side. The segments retrieved from different servers are not independent from each other and not aggregatable, they are divided into smaller non-decodable chunks spread over several servers. Our approach differs from [17], first by keeping client's side mechanism of existing single-server adaptive streaming solutions (thus being DASH-backward-compliant and technology-compliant), and second, by removing all dependency between sub-streams distributed on the servers (each sub-stream being decodable and playable by a standard video decoder). Moreover, the ability of our

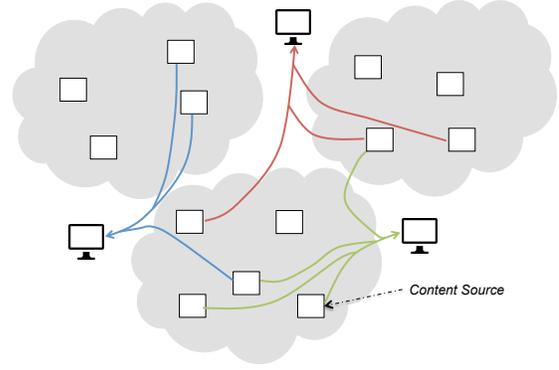


Fig. 1. MS-Stream solution overview

proposal to aggregate sub-streams is significantly improving the perceived quality. MS-Stream also includes a mechanism to address server's side constraints and to optimize service environment availability by adapting the type of descriptions served. This work in progress will be addressed in a future study. Last, in [18], an evolution of DASH is put forward using multiple servers assisted with Scalable Video Coding technique. Clients simultaneously request segment layers from several servers. Dependency between layers makes segment scheduling a complex task and, failing in retrieving the so called "base layer" will prevent the consumer from watching the video. MS-Stream does not have this limitation as each sub-stream, (i.e., *description*) is independent from the other and therefore provide the consumer with a smooth playback experience when descriptions are lost.

III. THE MULTIPLE-SOURCE STREAMING OVER HTTP APPROACH: MS-STREAM

We propose the Multiple-Source Streaming over HTTP protocol (MS-Stream) as an evolution of current adaptive streaming protocols, targeting consumer's perceived quality enhancement. Fig.1 depicts a high level overview of the MS-Stream solution, where any content source could contribute to streaming sessions, in accordance with their network connectivity type. The contributions of the paper are, first, the standard-compliant sub-stream generation scheme to be used by the MS-Stream protocol, and second, the MS-Stream protocol itself with its description adaptation, server-switching mechanism and its intra-description switching algorithm. The evaluation is presented in section IV.

A. The proposed sub-stream generation scheme

This section presents the sub-stream generation scheme utilized in our multiple-source streaming approach relying on the principles brought out by Multiple Description Coding (MDC) [4]. Each video is generated over multiple standard-compliant sub-streams, referred as descriptions. The number of descriptions, for a given stream, is calculated prior the delivery phase, based on (1) the sum of throughput upload capability of all available servers, (2) the servers' upload capacity cumulative distribution function, (3) the maximum video bitrate to be delivered to the consumers and (4) the number of consumers to be served simultaneously.



Fig. 2. Multiple-description generation process from H.264 content

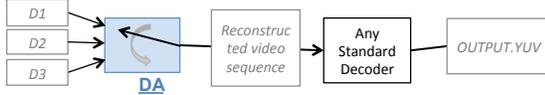


Fig. 3. Multiple-description decoding with the Description Aggregator (DA)

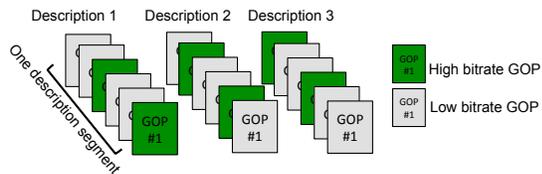


Fig. 4. Example of GOP repartition to create 3 different descriptions

Video quality increases with the number of descriptions merged; the original quality is obtained when all descriptions are used. Targeting ease of adoption of this sub-stream generation scheme, our contribution features: Video-codec standard compatibility, tunable redundancy, low additional encoding/decoding complexity, the possibility to create as many descriptions as needed, according to design decisions.

Video standard compatibility allows receivers to use a standard decoder module to decode descriptions. A low-complexity pre-decoding step is required to merge descriptions, thus preserving standard compatibility.

Descriptions are independent from each other; this is made possible by copying some common information (i.e., redundancy) into all descriptions. Non-dependency between flows provides high reliability in heterogeneous or unreliable networks, as any sub-stream can be lost without interrupting the streaming session. The more the redundancy, the greater the network bandwidth consumption overhead is. Consequently, and in order to match any specific scenario (e.g., from reliable sources with high throughput to volatile sources with low throughput), MS-Stream has embedded mechanisms to control the degree of redundancy based on servers availability, on network conditions and on the quality requested by the client.

Although most of the MDC approaches proposed in the literature [4] rely on two descriptions only, our solution can generate any number of descriptions, providing better bitrate scalability and resilience to outages and delay variations, as explained in [4]. As noted above, MS-Stream relies on the MDC principles. Four major MDC domains have been studied in the literature [4]: spatial, temporal, frequency, and compression. In our work, the focus has been placed on spatial and temporal domains since they allow codec standard-compliance. From our previous work [19], we propose a multiple-bitrate multiple-description coding strategy based on Group Of Pictures (GOP). Description creation is the result of a post-encoding step, which interleaves GOPs from two bitrate representations of the same content, as shown in Fig.2. Examples of GOP-based descriptions are depicted in Fig.4. Reconstructing the original content from the generated descriptions is a simple matter of data re-ordering, by selecting

the GOPs of higher size in the pool of available descriptions (Fig.3). In case not all descriptions are available for content reconstruction at client side, the content is still playable with a sub-optimal visual quality, due to some loss of information at the GOP level (inducing a lower visual quality on a GOP-duration basis).

B. The MS-Stream protocol and associated algorithms

The above-mentioned sub-stream generation scheme has given means to create independent descriptions to be placed into multiple source servers. This non-dependency between flows is a strong asset to ease the design of an efficient streaming protocol (compared to layered-based solution such as Scalable Video Codec). We propose an evolutionary streaming protocol: Multiple-Source Streaming over HTTP (MS-Stream), where client-side aspects are kept and a pragmatic usage of network bandwidth over multiple paths is suggested: (1) simultaneous content retrieval from several servers, (2) content adaptation, (3) server-switching mechanism and (4) a per-description "most available" server-selection algorithm.

1) *Multiple-server streaming protocol*: By retrieving several descriptions from several servers at a time, the probability to receive at least one stream is increased and a smooth content playback is assured (leading to a QoE increase). When all descriptions are retrieved, the highest video quality can be reconstructed and displayed. MS-Stream takes control decisions just before the download of each segment. Consider a scenario where the client is downloading high-bitrate 6 Mbps segment in good network conditions. The network bandwidth suddenly drops to 1 Mbps as the client has just started a new segment download. The segment will take 6 times the playout duration to download, depleting the available buffered content and possibly causing rebuffering. MS-Stream mitigates this problem by monitoring download progress and eventually abandoning a description segment download. Two metrics are used to define a timeout delay set on every request: buffered content and default segment playout duration. If the buffered content can provide content playout for less than twice the default segment playout duration, then the timeout delay is set to 90% of the chunk playout duration. Otherwise, no timer is set on segment request. In the event of all ongoing description segment requests (for a given reconstructed segment) lasting longer than the defined timeout, requests are kept alive until one description download is completed. Finally, if the available buffered content playout is less than half the default segment duration sequence and one description has already been received, the other complementary ongoing description downloads are abandoned in order to avoid rebuffering. This approach enables a smoother video playback on the consumer side, although with potential temporary sub-optimal displayed content quality.

2) *Description adaptation*: As the DASH (or other adaptive bitrate protocols) standard, our MS-Stream protocol has adaptation capabilities. Upon the reception of description segments, MS-Stream clients estimate the available bandwidth on the path used for a given description segment and, consequently,

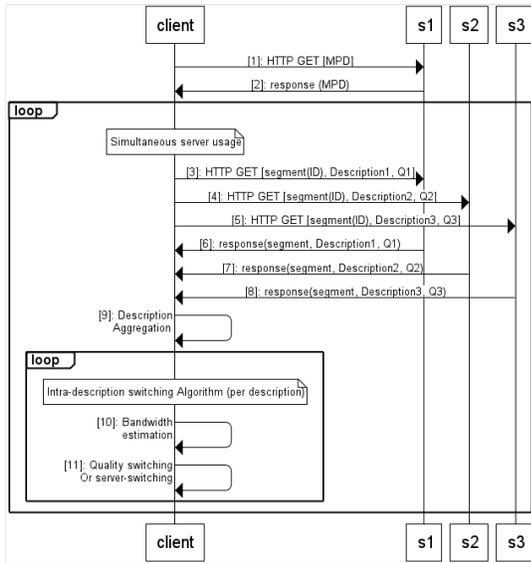


Fig. 5. Example of MS-Stream sequence diagram using 3 servers

can increase or decrease the bitrate representation of the next description segment to be downloaded. By creating several quality representations per description, the description adaptation features of MS-Stream addresses three different issues rising from MDC-based distributed streaming solutions: (1) minimizing negative QoE impact of description loss events, (2) minimizing bandwidth overhead coming from the multiple description coding approach and, (3) enhancing the maximum delivered content visual quality.

The first issue is directly linked to the network path quality used for the delivery of one description. Consider a client downloading a description segment on a poor condition network path (low throughput, high delay, high jitter, outages, packet loss, etc.), the probability to receive the latter segment outdated is high. MS-Stream mitigates this problem by letting two choices to clients: they can decide to retrieve the next description segment with a lower amount of redundancy (i.e., decreasing the segment bitrate, thus easing the delivery of the latter description), and/or they can decide to retrieve the next complementary descriptions segments (which are being retrieved from other servers) with an increased amount of redundancy, in order to compensate for the loss of the identified description.

The second issue being addressed by the MS-Stream description adaptation consists of minimizing the network bandwidth consumption overhead coming from multiple description coding. The overhead is defined as the percentage of data that transits on the network and that does not take part into the displayed content. We have $PlayedData + NonPlayedData = TransitingData$, therefore:

$$Overhead = 1 - \frac{PlayedData}{TransitingData} \quad (1)$$

with $PlayedData = \int_{t_{begin}}^{t_{end}} v_{bitrate}(x) dx$ with $v_{bitrate}(x)$ being the video played bitrate at time x , t_{begin} and t_{end} respectively being the start and end time of the video; and

$$TransitingData = \sum_{d=0}^D \sum_{s=0}^N \int_{t_{(s,d)start}}^{t_{(s,d)stop}} bitrate_{d,s}(x) dx$$

with D the number of delivered description per segment, N the number of segments in the video, $bitrate_{d,s}(x)$ the bitrate of segment s from description d , $t_{(s,d)start}$ and $t_{(s,d)stop}$ respectively being the download start and stop time of segment s from description d .

The global upload capacity of the considered streaming infrastructure is linearly decreased by the amount of redundancy added into each description. Since such overhead has a direct impact on the scale-up capacity of the considered streaming system to accommodate as many clients as possible, minimizing it is an essential issue. This issue is not directly linked to the consumer's QoE but rather to the server's side constraints, which, to its turn, impacts on consumer's QoE. Consequently, when a steady state is reached in a streaming session where description segments are always being received in time, MS-Stream clients are given the possibility to modify segment requests in order to lower the amount of redundancy per description (i.e., down to an almost null overhead).

The last issue is related to providing the optimal content quality while ensuring smooth video playback. When the observed throughput decreases, quality down-switching is performed by reducing the "high quality" contained into the delivered description segments and in order to decrease the bitrate of segments transiting on the network paths. Switching to a higher quality is performed when the observed throughput becomes $x\%$ greater than the next higher-quality's throughput. The value of x is determined by the streaming protocol provider in order to define more aggressive or more conservative consumption behaviors.

3) *Server switching mechanism*: The MS-Stream protocol also offers clients the opportunity to simply perform server-switching operations during streaming session. While relying on the smooth delivery of other descriptions, server switching is carried out by selecting another server for the delivery of the considered description with very small risks of buffer stalling. Two different cases are addressed by the proposed MS-Stream server adaptation mechanism. The first one is related to clients that are using servers for which the upload capacity becomes insufficient to deliver the requested descriptions in the desired quality. Switching to another server would allow such clients to eventually receive the desired description segments. The second one is related to clients that could only find suitable servers to retrieve description segments in sub-optimal quality (i.e. servers/paths that cannot deliver the best description quality according to description adaptation mechanisms). In such case, providing MS-Stream clients with the opportunity to perform server handover is a strong asset to avoid stagnating situations and to give clients the possibility to find servers and paths achieving higher throughput. The sequence diagram in Fig.5 depicts the content consumption process of MS-Stream client with its content adaptation and server-switching mechanisms. Implementation-wise, we defined an extension of DASH standard Media Presentation Description (MPD) file [2] in order to provide MS-Stream client with additional

information (available descriptions, available servers hosting the latter descriptions) and to assure backward compatibility with the DASH protocol.

4) *Intra-description switching algorithm*: In order to merge together both description adaptation and server-switching decisions, we propose the following decision algorithm 1. The algorithm's goal is to decide whether to perform description adaptation or server adaptation for one description at a time and how to do so (hence the name "intra-description switching"). The algorithm bootstraps with the available information from the MPD file that contains a list of servers associated with the hosted descriptions. The different variables influencing the intra-description switching algorithm are listed in Table I. The algorithm aims at, by order of priority, (i) keeping a server that delivers the requested description quality (line 4), (ii) increasing the requested description quality, when possible (line 5-6), (iii) switching to another server, if the client cannot obtain the optimal quality for a specific amount of requests (defined by the *stagnation period detection window* of M requests) (line 8-9); the *PerformServerHandover* directive is only performed if no other server handover is taking place at the same time for another description; and (iv) eventually lowering the requested description quality if and only if all available servers listed in the MPD file have been tested for the current quality (line 11-12-13). The last step of the algorithm is to re-compute the different variables for the next segment request (line 21): the timeout T value, the observation window defined by the N value and the stagnation period detection window defined by M . The *observation window* (N) is used to decide whether or not a server can properly stream the requested quality.

```

input:  $S, s$ 
1 while Streaming session is not over do
2   downloadsegment( $s, T, Q$ )
3   updateServerHistory()
4   if  $s$  successfully served the last  $N$  segments in
   quality  $Q$  then
5     if IncreaseQualityDecision() is true then
6        $Q = Q + 1$ 
7     else
8       if ( $Q < Q_{MAX}$  for the last  $M$  segments
       then
9          $s \leftarrow$  performServerHandover()
10    else
11      if quality  $Q$  was tested on all servers in  $S$  then
12        if  $Q \neq 0$  then
13           $Q \leftarrow Q - 1$ 
14        else
15           $s \leftarrow$  performServerHandover()
16    ( $T, N, M$ )  $\leftarrow$  RecomputeTNM( $Q, s$ )

```

Algorithm 1: Intra-Description switching algorithm

TABLE I
INTRA-DESCRIPTION SWITCHING VARIABLES

Symbol	Description	Initial Value
S	The set of available servers for a given streaming sessions	
s	A server in S	
Q_{MAX}	Maximum available description quality	
Q_{MIN}	Minimum available description quality	
Q	Requested description quality	Q_{MIN}
T	Request timeout value (seconds)	90% of chunk duration
ServerHistory	Array of server usage history	
N	Window of observation: number of consecutive requests used to determine the behavior of a server	1
M	Stagnation period detection window: number of consecutive requests used to determine if a client is blocked with a non-optimal server	5

IV. EVALUATION AND PERFORMANCE RESULTS

A. Empirical study - Test-bed

Our benchmark includes three implemented streaming applications: one MDC-streaming (3 available servers, using 3 servers simultaneously, without any kind of adaptation, each server hosting a different description); one MS-Stream-CA (3 available servers, using 3 servers simultaneously, each server hosting a different description) with Content bitrate Adaptation, as presented in section III-B2; and one MS-Stream-IS client (9 available servers, using 3 servers simultaneously, each group of 3 servers hosting a different description, as depicted in Fig.6) that includes the Intra-description Switching algorithm from section III-B4 (in the experiments, T , N and M were hardcoded to their initial values given in Table I).

Comparing our solution to existing state-of-the-art multiple-server adaptive streaming proposals (such as the ones presented in section II) is not simple since implementations are not always existing/available or are significantly complex to reproduce. To address this issue, we compare our solution to the full potential of both DASH (downloading one segment after the other) and MS-Stream in multi-source environments with ORACLE's clients. The ORACLE clients' goal is to maximize consumers' QoE by retrieving the highest content quality and minimizing video freezing events, as well as quality fluctuations. In addition, they know in advance all upcoming bandwidth variations and can always choose the best paths. The potential of DASH is represented by a DASH-ORACLE-3S client working with 3 servers, and by a DASH-ORACLE-9S client along with 9 available servers. The full potential of MS-Stream is defined by MS-Stream-CA-ORACLE and MS-Stream-IS-ORACLE clients, which feature the same characteristics as in the MS-Stream-CA and MS-Stream-IS clients. In this evaluation, we consider the perceived quality at the consumer's side, i.e., QoE. To this end, we have derived 5 criterias (each considered as essential for the QoE of video streaming services) to evaluate the seven aforementioned applications: (1) number of rebuffering events (Fig.7), (2) quality distribution throughout the streaming session (Fig.8 and Fig.9), (3) number of quality changes (Fig.10), (4) mean quality change amplitude (in kbps, available in Fig.11), (5) mean displayed content bitrate (Fig.12) representing data contributing to the rendered visual quality only. Finally, we also evaluated the percentage of data transiting on the network

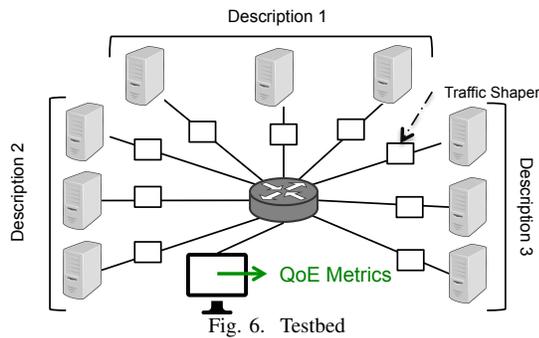


Fig. 6. Testbed

TABLE II
TESTBED APPLICATIONS

Applications	Number of available servers	Number of simultaneously used servers	Adaptation capabilities
MDC-Streaming	3	3	None
MS-Stream-CA	3	3	Description bitrate adaptation
MS-Stream-IS	9	3	Intra-description Switching Algorithm (Description bitrate adaptation + server switching)
MS-Stream-CA-ORACLE	3	3	Description bitrate adaptation
MS-Stream-IS-ORACLE	9	3	Description bitrate adaptation + server switching
DASH-ORACLE-3S	3	1	Bitrate adaptation + server switching
DASH-ORACLE-9S	9	1	Bitrate adaptation + server switching

that does not actually take part in the rendered video quality, defined as the data overhead from equation 1 (Fig.13). Table II sums up the evaluated solutions with their characteristics.

The evaluation has been performed with the 10-minute Big Buck Bunny movie[20], H.264 encoded at 6 different bitrates. Three independent descriptions were generated for each bitrate, and descriptions were sliced in 6-second segments (composed of 12 GoPs). Although each quality index has a specified average bitrate, chunks may have variable bitrate because of the varying nature of the movie. Each description segment is composed of 4 out of 12 GoPs in high bitrate and 8 out of 12 GoPs in low bitrate. Six different high bitrates are inserted into description: 6, 4, 3, 2, 1.5 and 1 Mbps. The redundant (low) bitrate is set to 200Kbps.

In the MDC-Streaming testbed, the client can only retrieve description with high bitrate at 6Mbps as it cannot perform any bitrate adaptation. We chose the 6Mbps quality in order to compare the behaviors of MDC-Streaming and MS-Stream-CA and MS-Stream-IS for the retrieval of the top quality. For MS-Stream-CA, MS-Stream-IS, MS-Stream-CA-ORACLE and MS-Stream-IS-ORACLE, clients can choose any of the available quality. The DASH-ORACLE dataset is composed of the same content quality as in the MS-Stream test-cases, although with regular H.264 encoded content.

The DASH Industry Forum provides benchmarks for various aspects of the DASH standard [21]. The benchmarks include twelve different network profiles (NPs). Profiles 1 to 6 have network bandwidths ranging from 1.5 to 5 Mbps, while profiles 7 to 12 have bandwidths ranging from 1 to 9 Mbps. Profile

TABLE III
NETWORK PROFILES

1	3	5	7	9	11
Mbps (ms;%)	Mbps (ms;%)	Mbps (ms;%)	Mbps (ms;%)	Mbps (ms;%)	Mbps (ms;%)
5.0 (38;0.09)	5.0 (13;0.81)	5.0 (11;1.00)			
4.0 (50;0.08)	4.0 (18;0.63)	4.0 (13;1.25)			
3.0 (75;0.06)	3.0 (28;0.44)	3.0 (15;1.50)			
2.0 (88;0.09)	2.0 (58;0.21)	2.0 (20;1.75)			
1.5 (100;0.12)	1.5 (200;0.03)	1.5 (25;2.00)			
2.0 (88;0.09)	2.0 (58;0.21)	2.0 (20;1.75)			
3.0 (75;0.06)	3.0 (28;0.44)	3.0 (15;1.50)			
4.0 (50;0.08)	4.0 (18;0.63)	4.0 (13;1.25)			
			9.0 (25;0.06)	9.0 (10;0.40)	9.0 (6;1.00)
			4.0 (50;0.07)	4.0 (50;0.08)	4.0 (13;1.25)
			2.0 (75;0.10)	2.0 (150;0.03)	2.0 (20;1.50)
			1.0 (100;0.16)	1.0 (200;0.07)	1.0 (25;2.00)
			2.0 (75;0.10)	2.0 (150;0.03)	2.0 (20;1.50)
			4.0 (50;0.07)	4.0 (50;0.08)	4.0 (13;1.25)

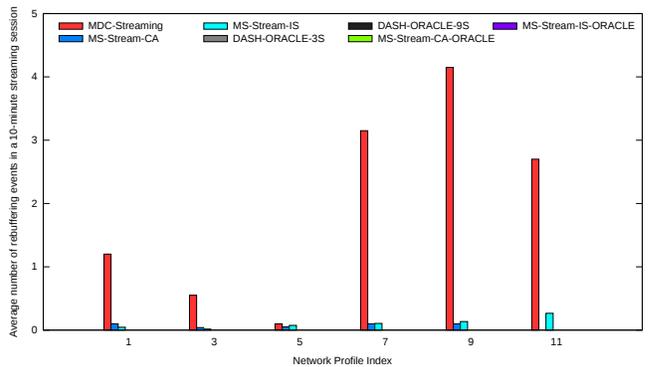


Fig. 7. Average number of rebuffering events per streaming session 1 spends 30s at each of 5, 4, 3, 2, 1.5, 2, 3 and 4 Mbps level respectively, then starts back at the top. Different latencies and packet loss are provided for each bandwidth. Table III shows the odd-numbered network profile characteristics. Even-numbered profiles are similar to the preceding odd-numbered profiles but start at the low bandwidth stage. For each experiment, a specific NP is associated to all channels between the client and the servers, as depicted in Fig.6. A random time offset was set to each assigned NP in order to represent bandwidth diversity and variability in the network. The MDC-Streaming, MS-Stream-CA, MS-Stream-IS applications were evaluated over all 6 NPs from Table III. Each experience (the 10-minute video streaming) was repeated 40 times and a total playback time of 120 hours was performed.

B. Empirical study - Results

1) *Rebuffering per session*: With DASH-ORACLE-3S and DASH-ORACLE-9S clients, rebuffering almost never took place. For the MDC-Streaming client, rebuffering occurred between 0.12 and 1.22 times per streaming session on NPs 1-3-5 and between 2.76 and 4.08 times per session on NPs 7-9-11. Thanks to their bitrate adaptation, MS-Stream-CA and MS-Stream-IS avoided buffer depletion to reduce the number of rebuffering to less than 0.22 times on all NPs.

2) *Content quality distribution per session*: MDC-Streaming displayed the top content quality for approximately 78% of the time on all NPs (see Fig.8). MS-Stream-CA performed a trade-off between content quality and network availability, thus providing smaller amount of top content quality compared to MDC-Streaming. Contrary to MDC-Streaming, which delivered the 200Kbps quality for the remaining time of streaming session, MS-Stream-CA used the set of available description representation to enhance the consumer's experience. By adding server-switching mechanisms to the protocol, MS-Stream-IS took benefit of a greater link diversity and eventually provided the 6Mbps top quality for 84% of the time in average on all NPs. In terms of quality distribution over the streaming session, MDC-Streaming, MS-Stream-CA and MS-Stream-IS performed better than the DASH-ORACLE-3S and DASH-ORACLE-9S, thanks to the simultaneous usage of servers. The maximum potential of MS-Stream-CA and MS-Stream-IS (represented by MS-Stream-CA-ORACLE and MS-Stream-IS-ORACLE respectively) showed that on all NPs, the 6Mbps quality was obtained for 99% of the time.

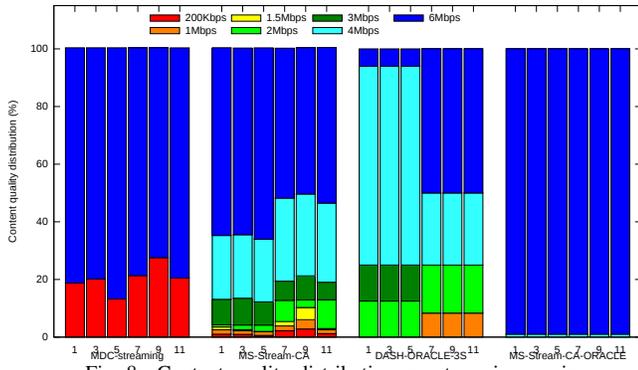


Fig. 8. Content quality distribution per streaming session

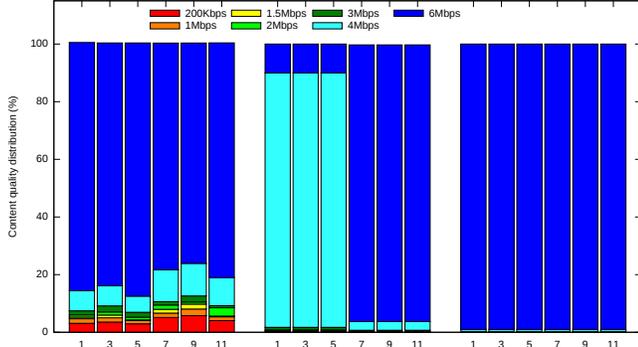


Fig. 9. Content quality distribution per streaming session

3) *Number of quality changes and mean amplitude of quality changes per session:* MDC-Streaming is the application that performed the worst in terms of quality changes: many quality changes (from 26 to 51 changes on all NPs according to Fig.10) with very high quality variation amplitudes (5800Kbps according to Fig.11) due to description segment download abandonment. For MS-Stream-CA, more quality changes occurred compared to MDC-Streaming (from 34 to 56 changes on all NPs) but with lower variation amplitudes, thus giving a better viewing experience to the video consumer. Regarding the average number of quality changes, MS-Stream-IS has in average 40.51% less changes than the MS-Stream-CA case on all NPs, thanks to the intra-description switching that attempts to retrieve content from other servers when the considered ones cannot provide sufficient throughput. Finally, for both DASH-ORACLE-3S and DASH-ORACLE-9S cases, quality changes are significantly less than for all applications on all NPs (less than 11 changes for DASH clients).

4) *Mean Bitrate per session:* DASH-ORACLE-3S reached a mean bitrate of approximately 3453Kbps on NPs 1-3-5 and 4475Kbps on NPs 7-9-11. In comparison, DASH-ORACLE-9S had 3 times the number of available servers of DASH-ORACLE-3S and respectively had a 21.66% and 29.12% mean bitrate increase on NPs 1-3-5 and 7-9-11. For the multiple-source clients, MDC-Streaming was able to perform better than DASH-ORACLE-3S on NPs 1-3-5 with an average increase of 43.32% and on NPs 7-9-11, 6.11%. Regarding the MS-Stream-CA application that performed bitrate adaptation in order to avoid rebuffering events, the obtained mean bitrate was not significantly different than MDC-Streaming. Finally, the MS-Stream-IS application was able to take benefit of

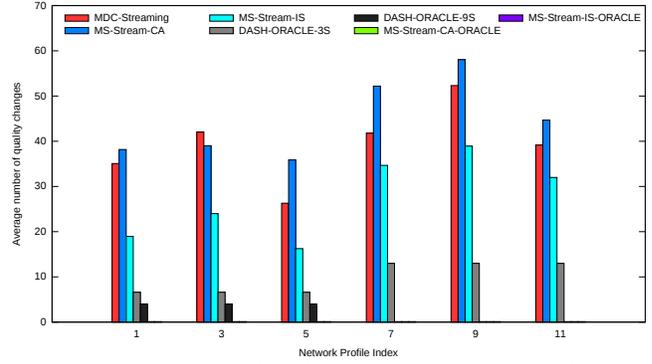


Fig. 10. Number of quality changes per streaming session

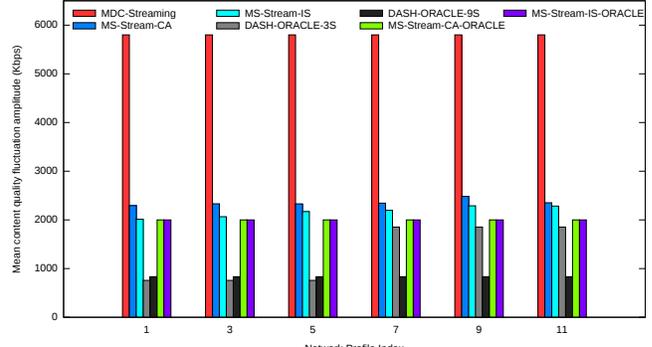


Fig. 11. Mean quality change amplitude per streaming session

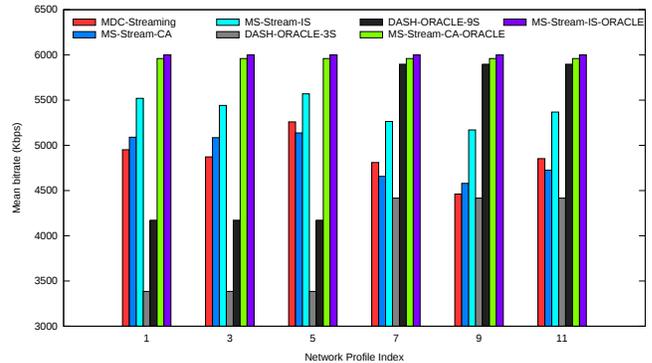


Fig. 12. Mean Bitrate per streaming session

path and source diversity and reached an average bitrate of 5552Kbps on NPs 1-3-5 and 5380Kbps on NPs 7-9-11, which represented a significant increase compared to MDC-Streaming. Compared to DASH-ORACLE-3S, MS-Stream-IS respectively performed 60.78% and 20.22% greater in terms of mean bitrate on NPs 1-3-5 and 7-9-11.

5) *Overhead:* In view of the overhead defined by equation 1, all multiple-description-based protocol presented an overhead lower than 8%. Interestingly, the MDC-Streaming application had the least amount of overhead compared to the other application. This is due to the fact that many description segment downloads were cancelled, and the redundancy from the delivered complementary description segments was used in order to display the content; thus decreasing the amount of data not taking part into the actual content displayed. For their parts, MS-Stream-CA and MS-Stream-IS have a greater overhead than MDC-Streaming with a slight increase on MS-Stream-CA as it retrieved the top qualities for in a smaller

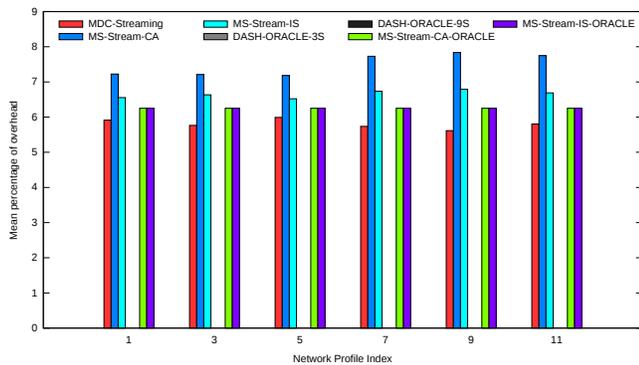


Fig. 13. Average percentage of overhead per streaming session proportion than MS-Stream-IS (6.57% of average overhead on all NPs), thus increasing the overhead coming from GOP redundancy. MS-Stream-IS’s overhead is very close to the one of the MS-Stream-IS-ORACLE that is 6.25%.

V. CONCLUSION

Within this paper, we presented MS-Stream: a solution to upgrade the legacy dynamic adaptive streaming protocols over HTTP with multiple-source support. The complex design and implementation issues resulting from server selection and segment scheduling problems were alleviated thanks to the proposed standard-compliant multiple-description strategy. Content bitrate adaptation and server-switching mechanisms were discussed and united into the proposed intra-description switching algorithm that performs a per-description adaptation strategy. We empirically demonstrated and compared the consumer’s QoE gain of MS-Stream based on an extensive collection of network profiles and against the optimal DASH solution, resulting in strong improvements for MS-Stream (up to 60% of mean bitrate increase). Finally, it must be noted that MS-Stream comes with a cost related with a necessary network bandwidth overhead, evaluated lower than 7% of the total traffic. The MS-Stream protocol has been implemented in a pragmatic vision relying on existing software guidelines and conforming to the H.264 and MPEG-DASH standards. A demonstration is available online at [6].

VI. FURTHER WORKS

The MS-Stream approach presented promising results in terms of consumer’s perceived quality gains. However, the issues of description placement in the network and content consumption demand prediction (that is done offline by the streaming providers) is not treated in our paper. Further work will include a study of a MS-Stream system incorporating online descriptions creation upon the demand and related to the specifications of clients. By enabling a new type of adaptation based on the reconstructed content representation, the expected gain is a better overall consumers’ QoE, especially concerning quality fluctuations. Thanks to this evolution, the issues resulting from description placement would no longer be and our solution would not add any complexity to the way streaming platform are designed and deployed. Finally, another study is being conducted on the optimal trade-off between increasing

the number of servers (i.e. the CPU and bandwidth overhead coming from MDC) versus the obtained QoE gain.

ACKNOWLEDGMENTS

This work is part of the DISEDAN project within the European CHIST-ERA Program, supported by the European Unions Future & Emerging Technologies scheme (FET).

REFERENCES

- [1] “Cisco Visual Networking Index, 2016,” <http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>.
- [2] “ISO/IEC 23009-1:2014 Dynamic adaptive streaming over HTTP.”
- [3] R. Pantos, “HTTP Live Streaming,” <https://tools.ietf.org/html/draft-pantos-http-live-streaming>.
- [4] M. Kazemi, S. Shirmohammadi, and K. H. Sadeghi, “A review of multiple description coding techniques for error-resilient video delivery,” *Multimedia Systems*, vol. 20, no. 3, p. 283309, 2013.
- [5] “Industry-Forum/dash.js,” github.com/dash-industry-forum/dash.js.
- [6] “MS-Stream Demonstration,” <http://www.labri.fr/perso/jbruneau/msstream/>.
- [7] M. Watson, “HTTP Adaptive Streaming in practice,” <http://web.cs.wpi.edu/~claypool/mmsys-2011/Keynote02.pdf>.
- [8] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, “Unreeling netflix: Understanding and improving multi-CDN movie delivery,” *2012 Proceedings IEEE INFOCOM*, 2012.
- [9] S. Akhshabi, A. C. Begen, and C. Dovrolis, “Experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP,” *Proceedings of the 2nd ACM conference on Multimedia systems - MMSys '11*, 2011.
- [10] “Microsoft Smooth Streaming,” <http://www.iis.net/learn/media/on-demand-smooth-streaming/smooth-streaming-technical-overview>.
- [11] “Adobe HTTP Dynamic Streaming,” <http://www.osmf.org/>.
- [12] C. Liu, I. Bouazizi, and M. Gabbouj, “Rate adaptation for adaptive HTTP streaming,” *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11*, 2011.
- [13] B. Zhou, J. Wang, Z. Zou, and J. Wen, “Bandwidth estimation and rate adaptation in HTTP streaming,” *2012 International Conference on Computing, Networking and Communications (ICNC)*, 2012.
- [14] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, “QDASH: a QoE-aware DASH system,” *Proceedings of the 3rd Multimedia Systems Conference - MMSys '12*, 2012.
- [15] L. D. Cicco, S. Mascolo, and V. Palmisano, “Feedback control for adaptive live video streaming,” *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11*, 2011.
- [16] G. Tian and Y. Liu, “Towards agile and smooth video adaptation in dynamic HTTP streaming,” *Proceedings of the 8th international conference on Emerging networking experiments and technologies - CoNEXT*, 2012.
- [17] S. Zhang, B. Li, and B. Li, “Presto: Towards fair and efficient HTTP adaptive streaming from multiple servers,” *2015 IEEE International Conference on Communications (ICC)*, 2015.
- [18] W. Pu, Z. Zou, and C. W. Chen, “Dynamic Adaptive Streaming over HTTP from Multiple Content Distribution Servers,” *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011.
- [19] J. Bruneau-Queyreix, D. Negru, J. Mongay Batalla, and E. Borcoci, “Multiple Description-DASH: Pragmatic video streaming maximizing End-Users’ Quality of Experience,” in *International Conference on Communications (ICC)*, 2016.
- [20] “Big Buck Bunny video,” <https://peach.blender.org>.
- [21] “Guidelines for implementation: DASH-AVC/264 Test cases and Vectors,” <http://dashif.org/wp-content/uploads/2015/04/dash-avc-264-test-vectors-v09-communityreview.pdf>.