Received: 30 March 2020 | Revised: 2 May 2020 | Accepted: 4 May 2020 | Published on: 15 June 2020

DOI: 10.1002/itl2.175

LETTER

WILEY

A DASH 360° immersive video streaming control system

Giuseppe Ribezzo | **Luca De Cicco** | **Vittorio Palmisano** | **Saverio Mascolo**

POLIBA, Politecnico di Bari, Italy

Correspondence
Luca De Cicco, POLIBA, Politecnico di Bari, Italy
Email: luca.decicco@poliba.it

Funding information
Ministero dello Sviluppo Economico, Grant/Award Number: F/050136/01/X32; Italian Ministry of Economic Development (MISE) through the CLIPS project (no. F/050136/01/X32).

This letter proposes a DASH-compliant video streaming control system for 360° immersive videos. The overall system is composed of two control algorithms which cooperate both to adapt the video bitrate to match the time-varying end-to-end network bandwidth and to select the most appropriate view of the panoramic scene in response to the varying point of view of the user. The proposed system has been implemented and an extensive experimental evaluation has been carried out in a realistic emulated network scenario to assess the obtainable performances in terms of visual quality and bitrate reduction.

KEYWORDS

augmented reality/virtual reality, experimental performance evaluation, immersive video streaming, MPEG-DASH

1 | INTRODUCTION AND BACKGROUND

Immersive video contents have been integrated in the services of several streaming platforms such as YouTube and Facebook due to the increasing popularity of Augmented-Virtual-Mixed Reality applications. Compared to traditional video streaming systems, the design of 360° video streaming platforms is a particularly challenging task, due to the extremely higher network bandwidth requirements.[1] To overcome this issue, the design of proper adaptive streaming techniques is a key research problem to solve. In such a context, the *MPEG Dynamic Adaptive Streaming over HTTP*(MPEG-DASH) is considered the de-facto standard in online video streaming. DASH systems allow clients to dynamically adapt the video bitrate to the time-varying network bandwidth.[2] At the server side, a content generation component encodes the original video at different bitrate *levels*or *representations*,[2] furtherly divided into segments of constant duration. The Media Presentation Description (MPD) contains a list of all video segments. At the client, a control algorithm dynamically selects the video level to be streamed at each segment download. The ultimate control goal of adaptive video streaming algorithms is to maximize the users perceived Quality of Experience (QoE) given the available bandwidth. Due to the special features of 360° videos which allow users to freely navigate the 360° scene, several content generation algorithms make use of viewport-adaptivity techniques[3] to reduce the overall bitrate of the streamed content. The basic idea of such techniques is to increase the quality in the video regions falling into the user's viewport while keeping a lower quality in regions that are likely to be out of the viewport. Nevertheless, a trade-off between server storage and complexity exists for 360° viewport-dependent DASH systems.[4]

In this letter, a complete DASH Immersive Delivery Solution is presented. The designed streaming system exploits the bitrate reduction technique presented in[3] as content generation algorithm. Moreover, a modular control algorithm has been conceived to provide both bitrate and viewport adaptivity. With respect to our previous work,[3] this paper presents a complete system for streaming omnidirectional contents. Due to space constraints we are not able to include in this paper an overview of the state of the art in the design of immersive video streaming solutions that is instead provided in.[3] The proposed DASH Immersive Delivery System has been evaluated under different emulated network scenarios. The obtained results show a 20% of bitrate reduction with a significant performance improvement in terms of video quality compared to the baseline case which does not employ our viewport adaptive scheme.

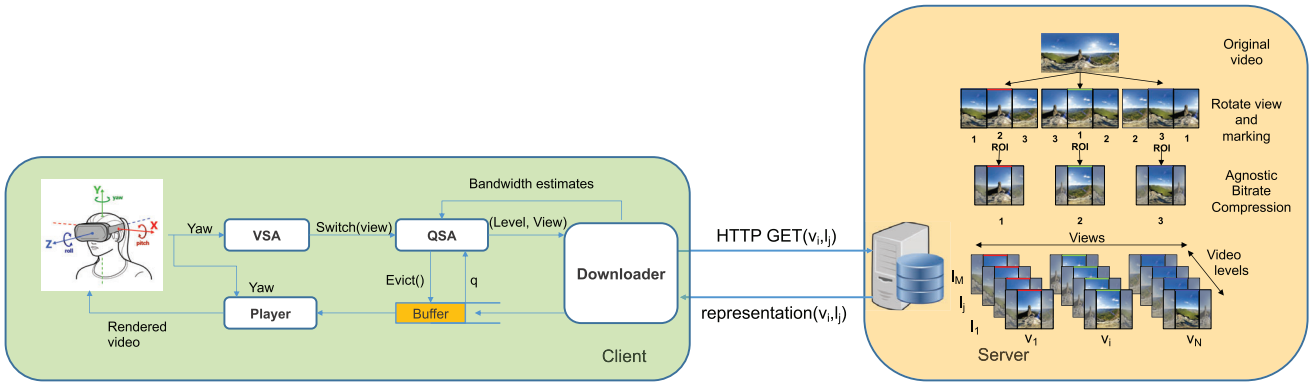

FIGURE 1 The proposed delivery system architecture

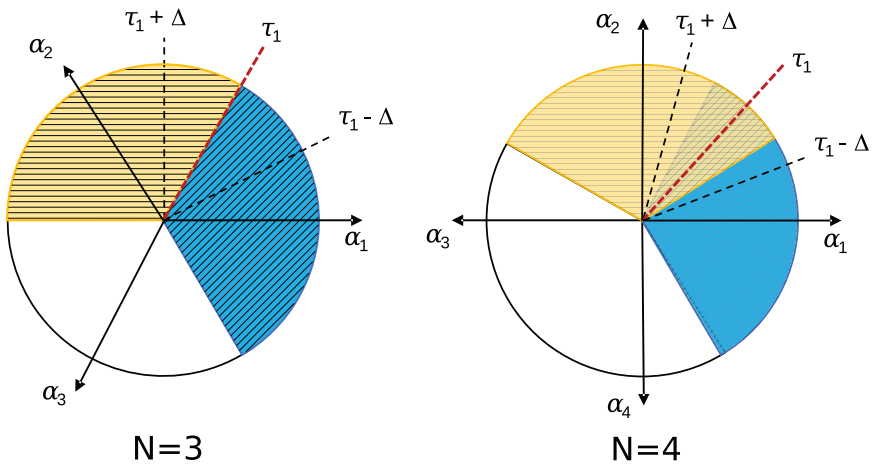

FIGURE 2 View selection algorithm in the case of $N = 3$ or $N = 4$ views

2 | THE PROPOSED IMMERSIVE VIDEO STREAMING SYSTEM

Figure 1 shows the overall architecture of the proposed delivery system which is composed of: a HTTP server hosting the video content generation system (Section 2.1); a player running at the client which manages the control logic and the rendering of the received video (Section 2.2).

2.1 | DASH-compliant server design

In DASH systems the server is composed by a content generation algorithm and a storage system exposing the video segments to be downloaded. In this letter, the content generation algorithm described in[3] has been implemented. To sum up, the Equirectangular Projection (ERP) has been considered the original uncompressed scene format. A particular RoI is identified as the*spherical lune* having a dihedral angle equal to 120°, centered at a particular yaw angle α_i. Since the video is represented in ERP format, rotation is applied to place the RoI at the center of the frame. The regions outside the RoI, namely the ones at its left (L) and its right (R), are divided into two spherical lunes of equal dihedral angle. Each spherical lune maps to a particular vertical strip of the video. The bitrate reduction is obtained downscaling the portions of the video outside the RoI. The idea is to generate from one ERP projected video a number N of versions, the *views*, that encode RoIs at different α_i. All the views form the *views set* $\mathcal{V} = \{v_1, \ldots, v_N\}$.Now each view $v_i \in \mathcal{V}$ is encoded into M video representations at different bitrates (and resolutions) l_j constituting the video*level set* $\mathcal{L} = \{l_1, \ldots, l_M\}$ with $l_j < l_{j+1}$. At the end of this procedure, the DASH Server stores and indexes a set of representations $\mathcal{R} = \mathcal{V} \times \mathcal{L}$ composed of $N \cdot M$ files.

2.2 | Viewport adaptive client

In DASH systems the client is composed by the *rendering engine* - able to render the downloaded 2D video chunks in a 3D virtual environment - and by the *control logic* - needed to dynamically select which video segment to download. In particular, such control logic requires two cooperating components: 2.2.1) a *quality selection algorithm* (QSA) that dynamically selects the video level $l(t) \in \mathcal{L}$ to avoid playback interruptions due to rebuffering while maximizing network channel utilization; 2.2.2) a *view selection algorithm* (VSA) which dynamically chooses the most suitable view representation $v(t) \in \mathcal{V}$ to be downloaded based on measurements provided by the HMD. The QSA acts similarly to classic DASH adaptive video streaming algorithms (see for instance[5]). The VSA, aiming at selecting the best view representation depending on the current user's head position, is a new component that immersive video delivery systems have to implement.

2.3 | Quality selection algorithm

Concerning the QSA, the ELASTIC[5] adaptive control algorithm is choosen. The control law employed by ELASTIC to dynamically select the video level is defined as follows:

$$l(t_k) = \begin{cases} l(t_{k-1}) & q_L \leq q(t_k) \leq q_H \\ Q\left(\frac{b(t_k)}{1-k_p e(t_k) - k_I e_I(t_k)}\right) & \text{otherwise} \end{cases} \tag{1}$$

where $q(t_k)$ is the playout buffer length measured in seconds, $b(t_k)$ is the available bandwidth estimated at the end of the download of the k-th segment and $Q(\cdot): x \mapsto l_i$ is a quantizer function mapping the bitrate x to the closest video level bitrate $l_i \in \mathcal{L}$ which is lower than x. The error $e(t_k)$ is given by

$$e(t_k) = \begin{cases} q_L - q(t_k) & q(t_k) < q_L, \\ q_H - q(t_k) & q(t_k) > q_H, \\ 0 & \text{otherwise.} \end{cases} ; \quad e_I(t_k) = \begin{cases} 0 & q_L \leq q(t_k) \leq q_H, \\ \sum_k (t_k - t_{k-1}) e(t_k) & \text{otherwise.} \end{cases} \tag{2}$$

Notice that $e_I(t_k)$ is the cumulative sum of the past values of the error $e(t_k)$. In a nutshell, the algorithm works as follows: as long as the playout buffer length stays inside the playout buffer hysteresis ($q_L \leq q(t_k) \leq q_H$) the video level is kept constant (1) to contain the amount of video level switches which is known to have an adverse effect on the QoE. When the playout buffer gets outside the hysteresis, the controller sets the video level according to (2). Notice that (2) aims at steering the playout buffer length $q(t)$ towards the hysteresis when the playout buffer length gets outside of it. Thus, if the available bandwidth is roughly constant, it turns out that the queue is confined in the hysteresis and the video level will switch between the two adjacent levels which are closer to the available bandwidth. An important consequence of this property is that ELASTIC ensures that the average video level bitrate matches the average available bandwidth.

2.4 | View selection algorithm

The design of the VSA for the selection of the view to be downloaded at a given time instant is described in the following. During the playback, the position of the head (ie, the current yaw angle $\alpha(t)$) is continuously measured and checked against the set of yaw angles α_i associated to the different views (recall α_i is the angular position of the RoI center of i-th view).

The VSA is designed to select the view that, based on the current position $\alpha(t)$, would provide to the user the largest high-quality area in the viewport. The approach is depicted in Figure 2 in the case of $N = 3$ or $N = 4$ views. The blue shaded area represents the RoI of the view v_0, whereas the yellow shaded area corresponds to the view v_1. For each view v_i a threshold τ_i is used to decide whether to switch to the view v_{i+1}. In particular, such thresholds are set as $\tau_i = (\alpha_{i+1} + \alpha_i)/2$.

Let us suppose the user is currently selecting view v_i. By employing such a setting for τ_i, it is very easy to check that when the user turns in a counterclockwise direction and $\alpha(t)$ surpasses τ_i, a switch to view $i+1$ is needed to guarantee

N	Strategy	Downsc. res. (px)	l_1(720p)	l_2 (1080p)	l_3 (2160p)
1	Baseline	1280	2.8 Mbps	5.2 Mbps	10 Mbps
3 or 6	*N*-480px	480	1.75 Mbps	3.25 Mbps	6.25 Mbps
	N-720px	720	2.1 Mbps	3.9 Mbps	7.5 Mbps

TABLE 1 Parameters used to encode the video levels and corresponding encoding bitrates

that the user enjoys the largest high-quality area in the viewport. In order to avoid unnecessary switches at the threshold boundary, an hysteresis centered on each τ_i is employed having an angular width equal to Δ degrees. The new optimal view is then chosen if the user crosses beyond such a threshold for more than K seconds, in order to limit the view switching frequency. When switching to a different view, to allow the user to perceive the improvement in visual quality within a reasonable time, a number of video frames stored in the buffer are evicted so that a configurable amount of playout buffer (named *safety margin*) is retained. This feature is extremely important because it allows to speed-up the visual improvement due to a view switch, while still allowing to avoid the occurrance of rebuffering events which are more likely to occur if the buffer occupancy is low.

3 | EXPERIMENTAL TESTBED AND SCENARIOS

To assess the effectiveness of the proposed system in a real scenario, the immersive video delivery system described in Section 2 has been implemented. On the one hand, a DASH Server (Debian Linux 9.12 workstation, 8GB RAM, Intel i7-4770 CPU @3.40GHz, running a 10.19.0 node version) has been set-up to implement the content generation algorithm described in Section 2.1, describing the video content in conformance to the specifications of the MPEG-DASH presentation format. On the other hand, the viewport-adaptive client designed in Section 2.2 has been developed as an HTML5 web player explicitly to be run on a common end-user laptop with Chromium web-browser. The web player makes use of standard technologies and open-source libraries to ensure compatibility with most modern browsers. The player exploits the WebGL-based open source library `THREE`* as the rendering engine. A specific mapping function (vertex shader) that properly associates the vertices of the mesh to the differently scaled strips of the video frame has been implemented to render the modified ERP produced by the content generation algorithm used at server side. The streaming engine has been built around the well-known open-source video streaming library Shaka-player†. Both the QSA and the VSA have been implemented in Shaka as plugins. The player has been also modified to introduce additional features, including the support for the adopted custom MPD and the ability to allow partial evictions from the video buffer. The QSA controller keeps track of both estimated bandwidth and playout buffer length choosing the best *bitrate level* to download, while the VSA chooses the best *view* in accordance to the head position of the user. The decision is taken during the download phase.

An extensive experimental campaign has been conducted to assess the performance gains offered by the proposed approach with respect to the usage of an adaptive streaming delivery system employing only one view identified in the following as the *baseline*approach. The classic dumb-bell network topology is employed. In particular, the client and the DASH server are connected through a bottleneck link with dynamically configurable capacity and latency though the `MahiMahi` tool.[6] To run the experiments, the 4 K video sequence "Elephants on the Brink (360 Video)"‡ was considered and the video player was instrumented to reproduce a realistic head movement according to the traces made available in.[7] The video has been encoded with different number of views ($N = 3$ or $N = 6$), downscaling resolutions (480px, 720px), and target video level bitrates as shown in Table 1. It is worth to note that in the case with $N = 6$ the resulting views contain ROIs partially overlapped.

The target bitrates for different parameter combinations have been chosen to ensure the same visual quality in the ROI region. This means that the visual quality of the ROI at a specific level l_i does not depend on the number of views and on the downscaling resolution of the regions falling outside the ROI. The 4 K video sequence has been encoded at 30 frames per second (fps) using the H.264 codec. The 4 K video sequence has been segmented using the MP4 container, with two different segment durations, respectively 1.6 and 3.2 seconds, and a group of picture (GOP) equal to the frame-rate multiplied by the segment duration, resulting in key-frames time-aligned across different levels and video segments. To evaluate the performance of the delivery systems considering different network conditions, four different mobile network traces (ATT-LTE-driving-2016, ATT-LTE-driving, TMobile-LTE-driving, Verizon-LTE-short) have been used, made publicly available by the MahiMahi suite§. For each trace, the average bandwidth along with the corresponding SD is

TABLE 2 Average bandwidth and SD for each considered network trace

	ATT-LTE-driving-2016	ATT-LTE-driving	TMobile-LTE-driving	Verizon-LTE-short
Average (DevStd) (Mbps)	5.2249 (0.0931)	7.4910 (0.1015)	9.0846 (0.3065)	4.6143 (0.3757)

TABLE 3 Average video segments bitrate reduction in the case of segments duration equal to 1.6 or 3.2 s

	3-480px	3-720px	6-480px	6-720px
1.6 s	21.9441 (2.7425) %	16.5618 (2.3358) %	24.1109 (5.9881) %	17.3631 (8.5262) %
3.2 s	11.7694 (4.6136) %	14.8192 (5.4837) %	18.9760 (6.3733) %	11.8218 (10.9257) %

shown in Table 2. Moreover, the performance of the system has been evaluated by considering two different safety margins, respectively equal to 5 or 3.2 seconds. Notice that draining the buffer to a lower margin may increase the likelihood of incurring in playback interruptions, but could also lead to an improvement of the visual quality in the viewport due to the higher responsiveness of the system reacting to user's head movements.

4 | RESULTS

In this section, the results obtained by employing the proposed approach integrated in a real-world DASH immersive adaptive streaming system are presented. Since the evaluation has not pointed out significant performance differences when employing a safety margin equal to 5 seconds, in the following only the results obtained in the case of the queue safety margin set equal to 3.2 seconds are reported.

Table 3 shows the average and SD of the reduction of the downloaded video segments bitrate with respect to the *baseline* case. It clearly shows that the reduction of segments bitrate is around 20% in each considered case: this is due to the lower target bitrate used to encode the video levels in the multiview case. To better clarify this result, let us make a concrete example. The difference between the target bitrate for the 2160p level respectively in the *baseline* case and the $N = 3$ views case is 3.75 Mbps, that is the maximum bitrate reduction reachable in the case the available bandwidth is large and the user movement are reduced (a low number of view switches is produced). However, when the user's head position changes and triggers a view-switch, the eviction of frames from the playout buffer can lead to downloading segments of the new view that were already downloaded for the previous view, thus lowering the obtainable bitrate reduction. This issue is slightly more evident in the case of larger segment duration (3.2 seconds) compared to the case of shorter segments (1.6 seconds).

Figure 3 shows four graphs grouped by each of the considered bandwidth traces. The graphs show the breakdown (expressed in percentage) of the visualized video levels for each considered video set parameter combination. A larger percentage of higher resolution levels corresponds to a better experienced visual quality. Figure 3 shows that the percentage of segments with higher level (2160p and 1080p) in the case of multiple views is larger than that obtained in the baseline case. This is due to the relative smaller target bitrate used to encode the video levels in the case of three or six view with respect to the baseline case. This result confirms that the proposed approach allows to improve the visual quality compared to the baseline case in any of the considered bandwidth traces and parameter settings. Notice that the best results are obtained in the case of $N = 3$ with a downscaled resolution equal to 480p followed by the case of $N = 6$ with downscaled resolution 480p. Overall it is not surprising that performances obtained for downscaled resolution equal to 480p are better compared to the 720p case. This is due to the fact that video levels corresponding to the 480p case can be encoded at a lower target bitrate compared to the 720p downscaled resolution case. Notice that we have measured a negligible performance increase using $N = 6$ instead of $N = 3$. In general, having a large number of views N leads to an increased frequency of view switches which then triggers more frequent buffer evictions and then an increased rebuffering probability. This result, together with the increased storage costs required for the additional views, shows that it is advisable to keep the number of views low. Moreover, in terms of rebuffering avoidance, the conceived streaming system has exhibited negligible rebuffering events (average around ∼0.5 events per video streaming session) independently from the employed strategy parameters. It must be stressed that the obtained results are roughly the same for the baseline case, indicating that the proposed solution does not worsen the performance in terms of rebuffering. This result is due to the robustness of the ELASTIC quality selection algorithm, known to provide very low rebuffering ratios.[5]

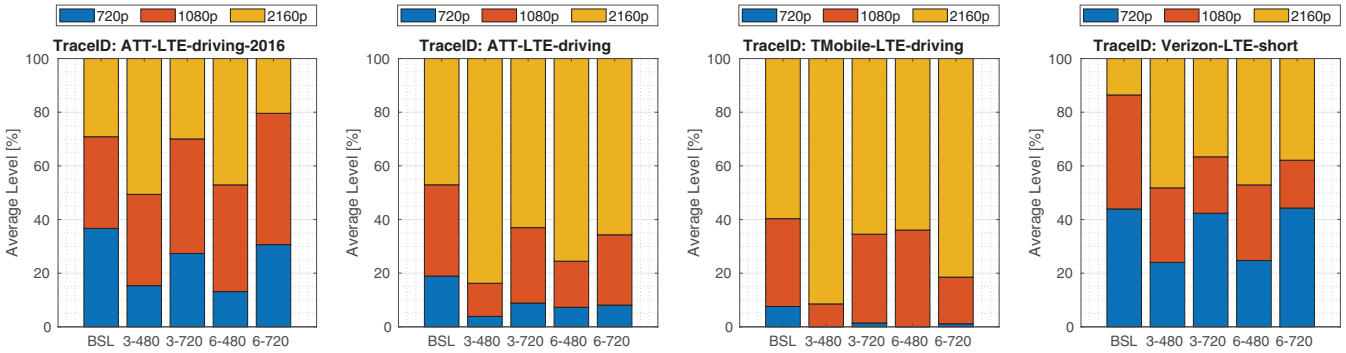

FIGURE 3 Breakdown of obtained video levels for each considered network trace

5 | CONCLUSIONS

In this letter, a complete DASH Immersive Delivery Solution has been presented. The system was designed to be immediately deployable using existing hardware platforms and Internet infrastructures. The experimental performance evaluation was carried out emulating a mobile network and shown that the proposed DASH System improves the obtained average visual quality while providing rebuffering ratios close to zero in each of the considered scenarios, which together concurs to improve the overall Quality of Experience for the streaming of 360° video.

ENDNOTES

* https://threejs.org
† https://github.com/google/shaka-player
‡ https://youtu.be/2bpICIClAIg
§ http://mahimahi.mit.edu/

ORCID

Giuseppe Ribezzo https://orcid.org/0000-0003-3057-0904
Luca De Cicco https://orcid.org/0000-0002-8900-175X

REFERENCES

1. Mangiante S, Klas G, Navon A, GuanHua Z, Ran J, Silva MD. VR is on the edge: how to deliver 360 videos in mobile networks. Paper presented at: ACM Workshop on Virtual Reality and Augmented Reality Network; 2017: 30–35.
2. ISO I. 23009-1. Information technology-Dynamic adaptive streaming over HTTP (DASH); 2014.
3. De Cicco L, Mascolo S, Palmisano V, Ribezzo G. Reducing the network bandwidth requirements for 360° immersive video streaming. *Internet Technology Letters*. 2019;2(4):e118.
4. Kammachi-Sreedhar K, Curcio ID. Omnidirectional video delivery with decoder instance reduction. *Internet Technology Letters*. 2019;2(1):e79.
5. De Cicco L, Caldaralo V, Palmisano V, Mascolo S. ELASTIC: a client-side controller for dynamic adaptive streaming over HTTP (DASH). Paper presented at 20th International Packet Video Workshop; 2013: 1–8
6. Netravali R, Sivaraman A, Das S, et al. Balakrishnan, Mahimahi: accurate record-and-replay for {HTTP}. Paper presented at USENIX Annual Technical Conference '15; 2015: 417–429.
7. Corbillon X, De Simone F, Simon G. 360-degree video head movement dataset. Paper presented at 8th ACM on Multimedia Systems Conference; 2017: 199–204.

How to cite this article: Ribezzo G, De Cicco L, Palmisano V, Mascolo S. A DASH 360° immersive video streaming control system. *Internet Technology Letters*. 2020;3:e175. https://doi.org/10.1002/itl2.175