A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP

Abdelhak Bentaleb, *Member, IEEE*, Bayan Taani, *Member, IEEE*, Ali C. Begen, *Senior Member, IEEE*, Christian Timmerer, *Senior Member, IEEE*, and Roger Zimmermann, *Senior Member, IEEE*

***Abstract*—In this survey, we present state-of-the-art bitrate adaptation algorithms for HTTP adaptive streaming (HAS). As a key distinction from other streaming approaches, the bitrate adaptation algorithms in HAS are chiefly executed at each client, *i.e.,* in a distributed manner. The objective of these algorithms is to ensure a high quality of experience (QoE) for viewers in the presence of bandwidth fluctuations due to factors like signal strength, network congestion, network reconvergence events, *etc.* While such fluctuations are common in public Internet, they can also occur in home networksor even managed networks where there is often admission control and QoS tools. Bitrate adaptation algorithms may take factors like bandwidth estimations, playback buffer fullness, device features, viewer preferences, and content features into account, albeit with different weights. Since the viewer's QoE needs to be determined in real-time during playback, objective metrics are generally used including number of buffer stalls, duration of startup delay, frequency and amount of quality oscillations, and video instability. By design, the standards for HAS do not mandate any particular adaptation algorithm, leaving it to system builders to innovate and implement their own method. This survey provides an overview of the different methods proposed over the last several years.**

***Index Terms*—Bitrate adaptation, HAS, DASH, adaptive video streaming, ABR schemes.**

I. Introduction

VIDEO delivery has evolved to constitute a major fraction of today's Internet traffic in the last decade thanks to advancements in network technologies, device capabilities, and audio-video compression schemes. Cisco reported in their annual Visual Networking Index that in 2016, 67% of the global Internet traffic was video, with a projection that it will reach 80% by 2021 [1]. This trend poses challenges in delivering videos with the best Quality of Experience (QoE) over today's Internet, which was originally designed for best-effort, non-real-time data transmission. Around 2005, an elegant yet simple video delivery paradigm was introduced by Move Networks, which quickly became popular due to its better features and cheaper deployment costs over progressive download and other proprietary streaming methods. This new paradigm, which we refer to as HTTP adaptive streaming (HAS), treated the media content like regular Web contentand delivered it in small pieces over HTTP protocol. HAS quickly became the dominant approach for video streaming due to its adoption by leading service and content providers. Video delivery over the public Internet is also referred to as over-the-top (OTT) video streaming, since the content or the streaming service provider usually differs from the network provider. The emergence of HAS and new, mostly mobile end-user devices with high processing and rendering capabilities played a key role in the growth of streaming video traffic.

In traditional non-HAS IP-based streaming, the client receives media that is typically *pushed* by a media server using either connection-oriented protocols such as the Real-time Messaging Protocol (RTMP/TCP) [2] or connectionless protocols such as the Real-time Transport Protocol (RTP/UDP) [3]. A common protocol to control the media servers in traditional streaming systems (as shown in Fig. 1a) is the Real-time Streaming Protocol (RTSP) [4]. RTSP is responsible for setting up a streaming session and keeping the state information during this session, but is not responsible for actual media delivery, which is the task for a protocol such as RTP. Based on the RTP Control Protocol (RTCP) reports sent by the client, the media server may perform rate adaptation and data delivery scheduling. These characteristics result in complex and expensive servers. Additional protocols or configurations are needed during the session establishment in case network address translation (NAT) devices and firewalls block the control or media traffic [5]. Despite implementing the same baseline protocol(s), media servers from different vendors may behave differently due to optional features or differences in implementation. Failovers due to a server fault often cause presentation glitches and are rarely seamless unless certain redundancy schemes are in place. These scalability and vendor dependency issues as well as high maintenance costs have resulted in deployment challenges for protocols like RTSP.

HAS uses HTTP as the application and TCP as the transport-layer protocol, as illustrated in Fig. 1b, and clients

Manuscript received December 21, 2017; revised May 19, 2018; accepted July 19, 2018. Date of publication August 3, 2018; date of current version February 22, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61472266, in part by the National University of Singapore (Suzhou) Research Institute, and in part by the Austrian Research Promotion Agency (FFG) under the Next Generation Video Streaming Project "PROMETHEUS." *(Corresponding author: Abdelhak Bentaleb.)*

A. Bentaleb, B. Taani, and R. Zimmermann are with the School of Computing, National University of Singapore, Singapore (e-mail: bentaleb@comp.nus.edu.sg; bayan@comp.nus.edu.sg; rogerz@comp.nus.edu.sg).

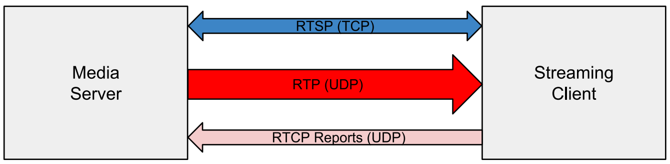
A. C. Begen is with Networked Media, Ozyegin University, Istanbul 34794, Turkey (e-mail: ali.begen@ozyegin.edu.tr).

C. Timmerer is with the Institute of Information Technology, Alpen-Adria Universität Klagenfurt, 9020 Klagenfurt, Austria, and also with Bitmovin Inc., San Francisco, CA 94105 USA (e-mail: christian.timmerer@itec.uni-klu.ac.at).

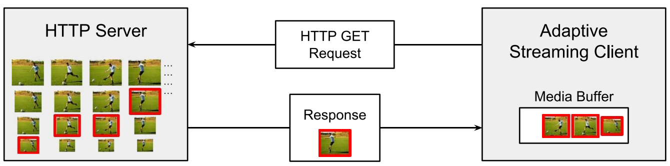
Digital Object Identifier 10.1109/COMST.2018.2862938

TABLE I
DIFFERENCES BETWEEN THE TRADITIONAL STREAMING AND HAS SYSTEMS

	Push-based Delivery (Traditional)	**Pull-based Delivery (HAS)**
Communication protocols	RTSP, RTP, UDP	HTTP, RTMPx, FTP
Adaptation logic runs at	Server side	Client side
Transmission data units	Packets	Media segments/chunks
Video monitoring and user tracking	RTCP for RTP transport	Currently proprietary, standardization is underway
Multicast support	Yes	No
Caching support	Protocol specific	Web caches as used for HTTP



(a) Traditional streaming with RTSP.



(b) HTTP adaptive streaming (HAS).

Fig. 1. Communication in traditional streaming and HAS systems.

pull the data from a standard HTTP server, which simply hosts the media content. HAS solutions employ dynamic adaptation with respect to varying network conditions to provide a seamless (or at least smoother) streaming experience. Once a media file (or stream) is ready from a source, it is prepared for streaming before it is published to a standard, off-the-shelf HTTP server. The original file/stream is partitioned into *segments* (also called *chunks*) of equi-length playback time. Multiple versions (also called representations) of each segment are generated that vary in bitrate/resolution/quality using an encoder or a transcoder (See Section II-A). Moreover, the server generates an index file, which is a manifest that lists the available representations including HTTP uniform resource locators (URLs) to identify the segments along with their availability times. During a typical HAS session, the client first receives the manifest that contains the metadata for video, audio, subtitles, and other features, then constantly measures certain parameters: available network bandwidth, buffer status, battery and CPU levels, *etc.* According to these parameters, the HAS client repeatedly fetches the most suitable next segment among the available representations from the server. Table I compares the main characteristics of the traditional streaming and HAS systems.

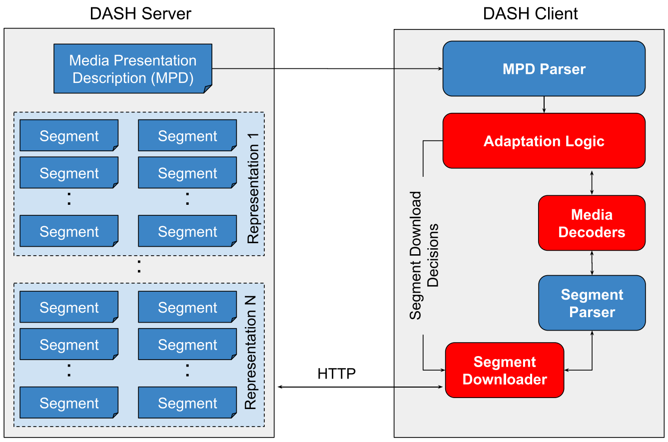
HAS is addressing several aspects that were major concerns in traditional streaming protocols [2]–[4]: *(1)* it uses HTTP to deliver video segments, which simplifies the traversal through NATs and firewalls [6]; *(2)* at the server side, it uses conventional Web servers or caches available within the networks of Internet Service Providers (ISPs) and Content Distribution Networks (CDNs); *(3)* a client requests and fetches each segment independently from others and maintains the playback session state, whereas the server is not required to maintain any state, hence, the client may download segments from different servers without impacting system scalability [7]; and *(4)* it does not require a persistent connection between the client and the server, which improves system scalability and reduces implementation and deployment costs.

Today, HAS accounts for the majority of Internet video traffic. It has reached mainstream due to commercial solutions such as Microsoft's Smooth Streaming [8], Apple's HTTP Live Streaming (HLS) [9], Adobe's HTTP Dynamic Streaming (HDS) [10], Akamai's HD [11] and several open-source solutions. To avoid fragmentation in the market, the Moving Picture Experts Group (MPEG) together with the 3rd Generation Partnership Project (3GPP) started working on HTTP streaming of MPEG media and HAS, respectively. These efforts eventually resulted in the standardization of Dynamic Adaptive Streaming over HTTP (DASH) [12]. Unlike proprietary solutions, DASH provides an open specification for adaptive streaming over HTTP and leaves the implementation of the adaptation logic to third parties as shown in Fig. 2a, where blue components are specified in the DASH standard, while red components are left unspecified or specified in other standards. The DASH server is essentially an HTTP server that hosts the media segments, which are typically two to ten seconds each, or could be as long as hours for the entire content duration in presentation time. Each segment is encoded at multiple bitrate levels and listed in the manifest termed Media Presentation Description (MPD, see Fig. 2b). The MPD is an XML document that provides an index for the available media segments at the server. At the client side, DASH implements the bitrate adaptation logic, which issues timed requests and downloads segments that are described in the MPD from the server using HTTP (partial) GET messages. During download, the DASH client estimates the available bandwidth in the network and uses information from the playback buffer to select a suitable bitrate for the next segment to be fetched. This behavior is called *bitrate switching*, where the client's goal is to fetch the highest-bitrate segments it can, while keeping sufficient data in the playback buffer to avoid video stalls and thus achieve a good QoE trade-off.

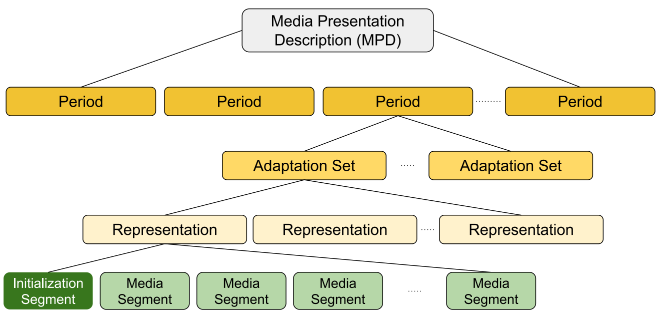
There are various implementations of DASH players.[1] For example, dash.js [13] is a JavaScript-based DASH client, which is the reference client from the DASH Industry Forum. Another JavaScript-based client is DASH-JS [14], which proposes a simple rate adaptation logic.

A recent survey [15] describes a range of bitrate adaptation (called also Adaptive BitRate (ABR)) schemes and techniques

[1] In this survey, the terms *player* and *client* are used interchangeably.



(a) DASH architecture components.



(b) Media Presentation Description (MPD) structure.

Fig. 2. Dynamic Adaptive Streaming over HTTP (DASH).

for DASH. The authors classified the schemes into three main categories: client-side, server-side and in-network approaches. They provided a general review of video traffic measurement methods and a set of characterization studies for well-known commercial streaming providers like Netflix, YouTube, and Akamai, and outlined several open research problems in the DASH streaming field. Our survey differs in terms of two key aspects: *(1)* a scheme classification is provided that is structured based on the unique features of the adaptation logics and *(2)* more schemes are examined and a detailed comparison table is provided.

Most state-of-the-art HAS solutions solely integrate the bitrate adaptation logic inside the HAS client, since it allows the client to select a bitrate level independently and avoids the requirement of having intelligent devices inside the network infrastructure. This represents a key reason why HAS solutions are used in OTT scenarios. Nevertheless, both industry and academia recommend using HAS systems in managed networks as well [16], [17]. For instance, a client may use feedback reported by a server or the network in bitrate adaptation to improve the overall QoE, or by using IP multicasting to simplify the video distribution in the context of connected TVs. In this survey, we present a classification of state-of-the-art bitrate adaptation schemes including features, pros, and cons. We classify the schemes into four main categories: client-based, server-based, network-assisted, and hybrid (See Fig. 3). The classification is based on the location of the bitrate adaptation logic within the system and which entities are involved.

The rest of this survey is organized as follows: Section II describes background information and definitions. Section III surveys the bitrate adaptation schemes. Comparisons between different schemes and a discussion are presented in Sections IV and V, respectively. Finally, Section VI provides concluding remarks.

II. Background and Definitions

A. Video Coding Standards

In an HAS system, a media file (or in the case of live video, a stream comprising chunks of audiovisual data) is encoded or transcoded into multiple representations. The most widely used video coding format is currently H.264, also known as MPEG-4 Advanced Video Coding (AVC) [18]. This video coding standard was introduced by MPEG in collaboration with the ITU-T Video Coding Experts Group (VCEG). A client requesting and downloading segments from possibly different representations (encoded at different bitrates) seamlessly concatenate these segments in its playback buffer. This results in a conforming bitstream that can be processed using a standard decoder. A common assumption is that each segment starts with an intra/key frame (*i.e.,* IDR-frame in AVC), in order for the decoder to process segments independently from each other. This may lead to coding inefficiencies for short segment durations [19].

Scalable Video Coding (SVC) has been introduced as an extension to AVC [20]. SVC enables splitting a video stream into multiple bitstreams or layers, where each one of them consists of subsets of video data. It recombines these bitstream subsets in order to additively increase the video quality. Typically, SVC allows the video stream to be split into three different dimensions of quality: temporal, spatial, and quality/Signal-to-Noise Ratio (SNR). In the temporal-based technique, the video is encoded at multiple frame rates for a given resolution. The base layer has the lowest frame rate, while enhancement layers increase the frame rate, which gradually improves quality. In the spatial-based technique, the video is encoded at multiple spatial resolutions for a given frame rate. In case of the SNR-based technique, the video is encoded at a single spatial resolution, and the enhancement layers improve quality, keeping the resolution constant.

The H.265 video codec (also known as High Efficiency Video Coding (HEVC)) was developed to provide approximately twice the encoding efficiency of AVC [21]. Similarly, as an extension to HEVC, Scalable High-efficiency Video Coding (SHVC) [22] was developed to support scalability. Conceptually similar to SVC, it adds extra scalability features such as bit-depth, color gamut, and hybrid scalability. In addition, it enhances coding-specific functionalities like Inter-Layer Prediction (ILP) (optionally encoding the base layer in AVC instead of HEVC), and the use of motion-constrained tiles. In both SVC and SHVC, the base layer is always backwards compatible with the non-scalable version of the encoder (AVC and HEVC, respectively), thus, only an AVC/HEVC decoder is needed.

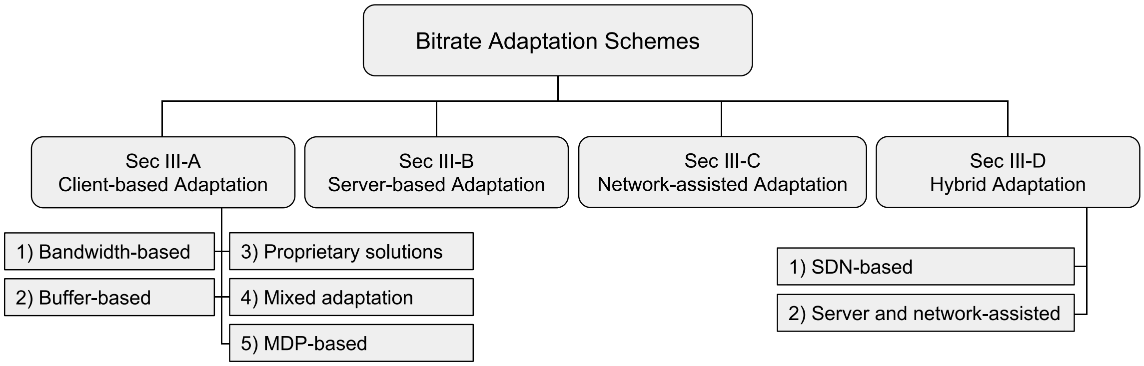


Fig. 3. HAS adaptation scheme classification.

Recently, MPEG and VCEG teamed up to work on Versatile Video Coding (VVC), aiming to provide almost twice the encoding efficiency of HEVC. VVC specifically targets applications and services using immersive and high-dynamic-range (HDR) videos. The new standard is expected to become available in 2020 [23].

Additionally, royalty free encoding formats such as VP9 and AV1 are increasingly used for HAS, and subject to various evaluations. For example, open-source implementations of AVC, HEVC and VP9 have been evaluated in large-scale video-on-demand environments [24].

B. Common Problems in HTTP Adaptive Streaming

While moving from a server-push to a client-pull model has clear benefits, HAS still faces challenges. Known issues relate to the heterogeneous nature of networks, the increasing number of users, and the growing demand of high-quality content. We describe four main problems that can affect HAS systems: *(1)* multi-client competition and stability issues, *(2)* consistent-quality streaming, *(3)* QoE optimization and measurement, and *(4)* inter-destination multimedia synchronization.

1) Multi-Client Competition/Stability Issues: Seufert *et al.* [25] have shown that using a centralized management controller can enhance the overall video quality, while improving the viewer QoE. In that regard, a robust HAS scheme should achieve three main objectives:

- *Stability:* HAS clients should avoid frequent bitrate switching, which leads to quality oscillations and video stalls, which in turn can negatively affect QoE.
- *Fairness:* Multiple HAS clients competing for available bandwidth should equally share network resources based on viewer-, content-, and device characteristics. The fairness desired here does not often result in bandwidth-fairness.
- *High Utilization:* While the clients attempt to be stable and fair, network resources should be used as efficiently as possible.

A streaming session in general consists of two states, the buffer-filling state and the steady state [26]. The buffer-filling state aims to fill the playback buffer and reach a certain threshold where the playback can be initiated or resumed. In this state, the client requests the next segment as soon as the previous chunk is fully downloaded (See Fig. 4). After the playback buffer level reaches a target threshold (*e.g.,* 30 seconds, however, note that this threshold varies among different bitrate adaptation schemes or could be increased or decreased based on the expected conditions), the client enters the steady state. The objective during the steady state is to keep the buffer level above a minimum threshold despite bandwidth fluctuations or interruptions, in order to avoid buffer underrun or stall events. The steady state consists of two activity periods referred to as ON and OFF. Fundamentally, an HAS client requests a segment every T_s time units, where T_s represents the content time duration of each segment, and the sum of ON and OFF period durations equals T_s. During the ON period, the HAS client downloads the current segment and notes the achieved throughput value that will be later used in selecting the appropriate bitrate for future segments. After that, the client temporarily becomes idle in the OFF period (See Fig. 4).

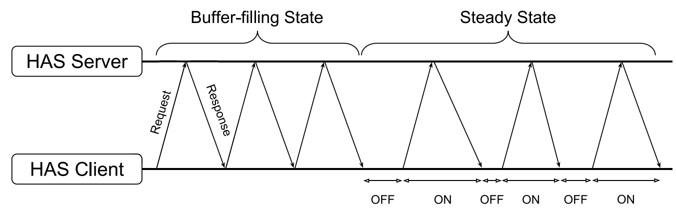
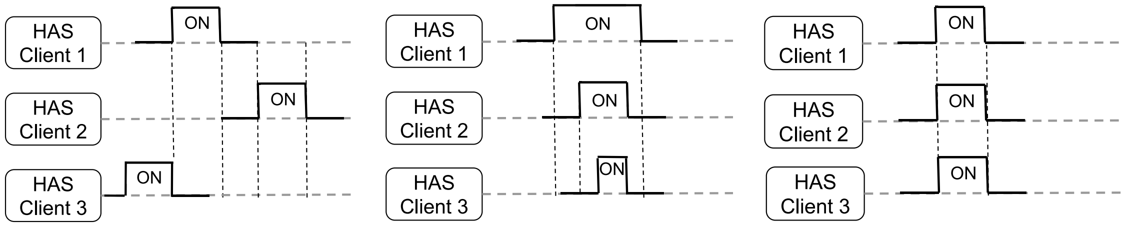


Fig. 4. HAS video streaming session states.

As shown in Fig. 5, when a set of HAS clients competes for the available bandwidth, the per-segment activity periods (ON, OFF) of the steady state differ from client to client. Depending on the amount of overlap of the ON periods, the clients may at times considerably overestimate the available bandwidth. This potentially causes video instability, quality oscillations, bitrate switches, buffer underruns, unfairness and underutilization, which are collectively referred to as HAS stability issues.

Consider, for example, three HAS clients that share a bottleneck link. Suppose that these three clients have reached the steady state and they request a new segment every T_s time units. As illustrated in Fig. 5a, if the ON periods of these clients do not overlap during the current segment download, each client will overestimate the available bandwidth. This



(a) Unsynchronized downloads. (b) Overlapping downloads. (c) Synchronized downloads.

Fig. 5. Illustration of the main cause of HAS stability issues because of different segment download patterns.

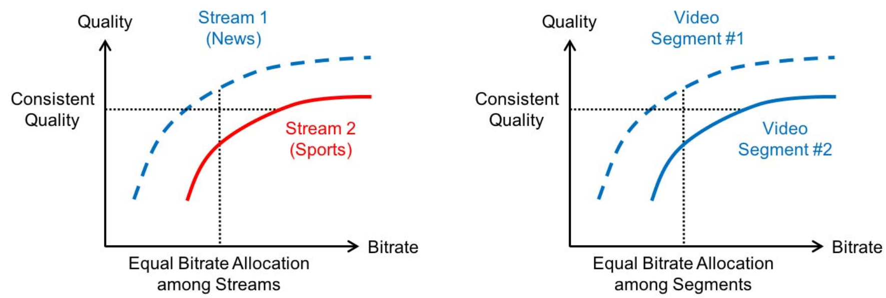


Fig. 6. Different inter-stream (on the left) and intra-stream (on the right) scene complexities lead to different display qualities at the same encoding bitrate or vice versa.

would not be the case if the ON periods were partially (Fig. 5b) or fully (Fig. 5c) overlapping. Many HAS bandwidth estimation algorithms use the current segment download speed as an input. Non-overlapping ON periods lead to overestimating the fair share of the bandwidth, and thus, clients incorrectly select a higher encoding bitrate for the next segment. Downloading the next segment, which has a higher encoding bitrate, will take longer, which will cause the initially non-overlapping ON periods to eventually start overlapping. As the amount of overlap increases, the clients will have lower bandwidth estimations and start selecting segments that have a lower encoding bitrate. These segments will take less time to download, causing the amount of overlap among the ON periods to procedurally shorten, until the process reverts to its initial situation. This cycle repeats itself, causing periodic up- and downshifts in the selected bitrates, leading to unstable video quality, unfairness, and underutilization [26]–[28].

2) Consistent-Quality Streaming: Research studies in the field of video quality analysis [29], [30] confirm that the correlation between video bitrate and its perceptual quality is non-linear. Additionally, different video content types have unique characteristics, *e.g.,* high and low-motion scenes, which result in different qualities.

In the context of HAS, even if the available bandwidth stays constant, the delivered video quality may still vary, as illustrated in Fig. 6, due to unequal video scene complexity across content: inter-stream and intra-stream differences. Fig. 7 depicts the non-linear relationship between bitrate and the Structural SIMilarity plus (SSIMplus) [31] perceptual quality. Generally speaking, it is preferred to stream video with a consistent quality than at a consistent bitrate [32], [33], leading to a reduction in perceptual quality oscillations.

3) QoE Optimization and Measurement: The changing conditions of best-effort networks introduce numerous problems

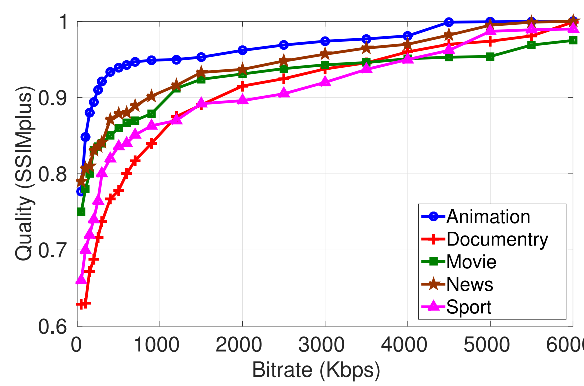


Fig. 7. Illustration of quality versus bitrate trade-off.

in delivering multimedia content to viewers. In traditional non-adaptive streaming, the client streams a video that is typically available in one bitrate at the server side. If the network conditions worsen, the download rate may fall below the playback rate, which leads to buffer depletion and discontinuous playback. With HAS, streamed videos show less buffering and higher bandwidth utilization compared to traditional streaming, since the video segments are transcoded into different bitrate levels, and segments are downloaded based on the current network conditions and the playout buffer level. Fig. 8 illustrates the application control loop of a typical HAS client. This survey focuses on reviewing the adaptation algorithms, *i.e.,* the part responsible for selecting the next segment(s) to download. The application control loop also interacts with a lower-layer control loop (in this case TCP congestion control), which can play a key role in determining the viewer QoE.

In a recent survey by Seufert *et al.* [25], factors influencing QoE are categorized as (*a*) perceptual, directly perceived by the viewer, and (*b*) technical, indirectly affecting the

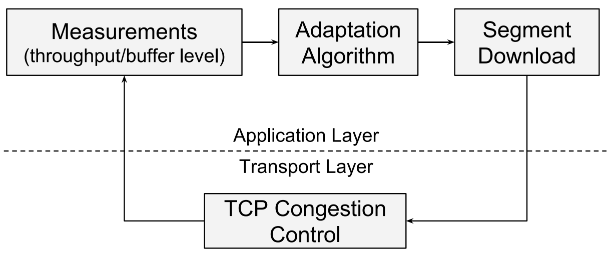


Fig. 8. The application control loop in a typical HAS client.

QoE. Perceptual factors include the video image quality, initial delay, stalling duration and frequency, as well as quality switching amplitude and frequency. The impact of each of these factors differs depending on the users subjectivity. Several studies have shown that most users consider initial delays less critical than stalling [34], [35], that longer stalling periods decrease the perceived quality [36], and that frequent changes in video quality have a negative impact on the QoE [37]–[39]. The technical factors that influence QoE are the algorithms, parameters, and hardware/software used in the video streaming system. Specifically, such factors include encoding parameters, video qualities and segment sizes at the server side, the adaptation logic, device capabilities and content type at the client side, as well as the adaptation parameters and the type of environment that the client resides in. All of these factors are challenges to be taken into account for the best trade-off between conflicting goals (*e.g.,* less stalling vs. high encoding bitrate) in order to achieve viewer satisfaction.

One major challenge regarding video streaming is the lack of a unified quantitative approach to measure the QoE. Existing HAS solutions in industry and academia assess their QoE based on three different metrics: *(1)* Objective metrics, such as Peak Signal-to-Noise Ratio (PSNR) [40], [41], Structural SIMilarity (SSIM and SSIMplus) [31], [42], Perceived Video Quality (PVQ) [43], and Statistically Indifferent Quality Variation (SIQV) [44]; *(2)* Subjective metrics, such as Mean Opinion Score (MOS); or *(3)* Quality-of-Service (QoS)-derived metrics such as the startup delay, average video bitrate, quality switches and rebuffering events. Achieving high QoE is difficult because trying to optimize each metric may result in conflicts. The complex relationship between these measures and the interplay between the adaptation logic with other application and network-layer decisions can significantly affect the QoE. Balachandran *et al.* [45] address these issues and propose a data-driven approach that uses machine-learning to build a QoE prediction model. They showed that it could enhance the user engagement when applied in a CDN.

4) Inter-Destination Multimedia Synchronization: The ever-growing development of social multimedia sites is changing the way people share content. Apart from online gaming, photo sharing, and instant messaging, online communities are drifting towards watching online videos together in a synchronized manner. Having multiple streaming clients distributed in different geographical locations poses challenges in delivering video content simultaneously, while keeping the playback state of each client the same (playing, paused). Moreover, it becomes more challenging for HAS streams to synchronize, since each client adaptively streams depending on their current network conditions. This problem is called Inter-Destination Multimedia Synchronization (IDMS). Typically, IDMS solutions involve a master node (either a dedicated master or a peer among the streaming clients in a session) to which clients synchronize their playout to. One of the earliest papers in this field was published by Montagud *et al.* [46], in which the authors discuss use cases where IDMS and its schemes is essential. Rainer *et al.* [47], [48] proposed an IDMS architecture for DASH by using a distributed control scheme (DCS) where peers can communicate and negotiate a reference playback timestamp in each session. The MPD file was altered to include IDMS session objects that enabled session management. In another work [49], Rainer *et al.* provided a crowdsourced subjective evaluation to find an asynchronism threshold at which QoE was not significantly affected. They found that an asynchronism level of 400 ms was acceptable compared to the synchronous reference case. Synchronization in IDMS systems is crucial to the QoE. Dedicated QoE models have to be developed that take the visual quality, user engagement and the synchronization accuracy into account.

After describing the various factors that affect Internet video streaming systems, we will now continue with a survey of the existing bitrate adaptation schemes.

III. Bitrate Adaptation Schemes

We classify bitrate adaptation schemes based on the entity of the system where the logic is implemented:

- Client-based adaptation (Section III-A),
- Server-based adaptation (Section III-B),
- Network-assisted adaptation (Section III-C), taking into account explicit information from within the network, and
- Hybrid adaptation (Section III-D), using information from any combination of the client, server(s), and network.

The taxonomy graph in Fig. 3 illustrates our classification of bitrate adaptation schemes.

A. Client-Based Adaptation

In relevant literature, most of the proposed bitrate adaptation schemes reside at the client side, according to the specifications in the DASH standard [50]. These schemes try to adapt to bandwidth variations by switching to an appropriate video bitrate according to one or more metrics such as the available bandwidth, playback buffer size, *etc.* Fig. 9 shows a simple model of a client-based adaptation. The client uses one or more metrics as input for its bitrate selection algorithm in order to choose the appropriate bitrate level for the next segment to be downloaded. These algorithms try to avoid streaming problems like video instability, quality oscillations, and buffer starvation, while improving viewer QoE. They strive to achieve *(i)* minimal rebuffering events when the playback buffer depletes, *(ii)* minimal startup delay especially in case

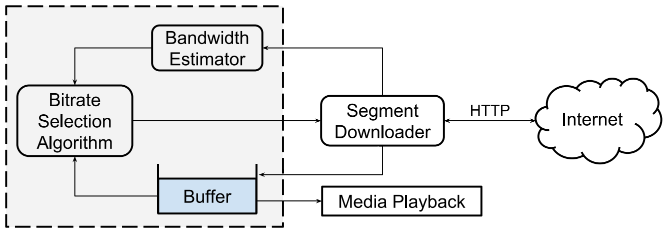


Fig. 9. Client-based bitrate adaptation.

of live video streaming, *(iii)* a high overall playback bitrate level with respect to network resources, and *(iv)* minimal video quality oscillations, which occur due to frequent switching.

We further organize the client-based bitrate adaption into five classes: *(1)* available bandwidth-based (Section III-A1), *(2)* playback buffer-based (Section III-A2), *(3)* proprietary solutions (Section III-A3), *(4)* mixed (Section III-A4), and *(5)* Markov Decision Process (MDP)-based (Section III-A5).

1) Available Bandwidth-Based Adaptation: In this intuitive type of scheme, the client makes its representation decisions based on the measured available network bandwidth, which is usually calculated as the size of the fetched segment(s) divided by the transfer time. Liu *et al.* [51] proposed a bitrate adaptation algorithm that tries to detect bandwidth fluctuations and congestion using a smoothed network throughput based on the segment fetch time (SFT), which measures the time starting from sending the HTTP GET request to receiving the last byte of the segment. Later, the authors extended their work in [52] to include both sequential and parallel segment fetching methods in CDNs, by using a metric that compares the expected segment fetch time (ESFT) with the measured SFT to determine if the selected segment bitrate matches the network capacity. A similar approach was employed by Rainer *et al.* [14] where the bandwidth estimated for the next segment was calculated based on the bitrate observed for the last segment downloaded and the estimated throughput that was calculated during the previous estimation. The initialization was based on the bandwidth measured when downloading the MPD.

Probe AND Adapt (PANDA) [53] estimates the available bandwidth accurately and tries to eliminate the ON-OFF steady state issue as well as reduce bitrate oscillations when multiple clients share the same bottleneck link. The video adaptation framework for DASH clients in LTE networks, piStream [54], enables clients to estimate the available bandwidth based on a resource monitor module that acts as a physical-layer daemon. Andelin *et al.* [55] integrated SVC with DASH by proposing an algorithm that prefetches base layers of future segments or downloads enhancement layers for existing segments using a bandwidth-sloping-based heuristic.

In live video streaming, the nature of the live experience puts stringent constraints on the delay. DASH to Mobile (DASH2M) [56] by Xiao *et al.*, is a strategy designed for mobile streaming clients using HTTP/2 server push and stream termination properties with the goal of enhancing the QoE as well as reducing the battery consumption of the client. An extension of the authors' previous work [57], the adaptive *k*-push scheme proposes to increase/decrease *k* according to a bandwidth increase/decrease while keeping in mind the overall power consumption in a push cycle. In the same context, Miller *et al.* [58] proposed a low-latency prediction based bitrate adaptation scheme over wireless access links termed LOw-LatencY Prediction-based adaPtation (LOLYPOP), which leverages TCP throughput predictions on multiple time scales (*i.e.,* 1 to 10 seconds) to achieve low latency and improve viewer QoE.

For the specific case of mobile clients that are in motion, the network conditions are more fluctuating with respect to location and time. Several studies deploy a bandwidth lookup service in a real-life mobile network in order to guide the bandwidth estimation among the mobile clients [59]–[63]. However, these frameworks take a spatial point of view of bandwidth fluctuations and pay little attention to the temporal factor. GeoStream [64] addresses this issue and introduces the use of geostatistics to estimate future bandwidth in unknown locations.

In general, available bandwidth-based adaptation suffers from poor QoE due to a lack of a reliable bandwidth estimation methods, which results in frequent buffer underruns.

2) Playback Buffer-Based Adaptation: In this type of scheme, the client uses the playout buffer occupancy as a criterion to select the next segment bitrate during video playback.

Mueller *et al.* [65] were motivated by the limitation of bandwidth-based adaptation when multiple clients competed for the available bandwidth, specifically in the presence of cache servers. Therefore, the authors proposed a buffer-based bitrate adaptation scheme that combines the buffer size with a tool-set of client metrics for accurate rate selection and smooth switching. Huang *et al.* [66] proposed a set of buffer-based rate selection algorithms, named BBA that aim to maximize the average video quality and avoid unnecessary rebuffering events. However, BBA suffers from QoE degradation during long-term bandwidth fluctuations. Buffer Occupancy based Lyapunov Algorithm (BOLA) [67], on the other hand, is an online control algorithm that treats bitrate adaptation as a utility maximization problem. This utility is associated with the average bitrate and rebuffering time, while adapting to network changes to account for better QoE. The authors provide strong theoretical proof that it is near optimal, design a QoE model that incorporates both the average playback quality and the rebuffering time, and empirically show its efficiency using various network traces. BOLA is the buffer-based algorithm that is implemented and available in the dash.js player.

Sieber *et al.* [68], introduced an SVC-based adaptation algorithm called Bandwidth Independent Efficient Buffering (BIEB). BIEB maximizes video quality based on SVC priority while reducing the number of quality oscillations and avoiding stalls and frequent bitrate switching. BIEB maintains a stable buffer occupancy before increasing the quality (enhancement layers). However, BIEB does not take bitrate switches or stalls in the QoE model during peak times when dynamic cross traffic occurs in the network into consideration.

The decision by these algorithms which bitrate to select largely depends on factors such as estimated network throughput, buffer occupancy, and buffer capacity. Yet, these algorithms are not informed by a fundamental relationship between these factors and the chosen bitrate. Thus, they do not work consistently in all scenarios. To address this issue, Yadav *et al.* [69] modeled a DASH client as an M/D/1/K queue referred to as a QUEuing Theory approach to DASH Rate Adaptation (QUETRA), which allowed them to calculate the expected buffer occupancy given a bitrate choice, network throughput, and buffer capacity. Using this model, the authors proposed a simple rate adaptation algorithm and evaluated QUETRA under a diverse set of scenarios. They found that despite its simplicity, QUETRA led to better QoE than the existing algorithms.

In general, buffer-based adaptation schemes suffer from many limitations including low overall QoE and instability issues, especially in the case of long-term bandwidth fluctuations. SVC-based approaches also have limitations related to the complexity of SVC encoding and decoding, processing resources and overhead. Some alternative solutions have tried to tackle these issues using multiple SVC streams, hierarchical encoding with a small number of enhancement layers, and encoding overhead [70].

3) Proprietary Solutions: In the past, we witnessed many proprietary adaptive streaming solutions and player implementations such as *Microsoft's Smooth Streaming* (MSS) [8], *Apple's HTTP Live Streaming* (HLS) [9], *Adobe's HTTP Dynamic Streaming* (HDS), and *Open Source Media Framework* (OSMF) [10]. These solutions use different metrics in their bitrate adaptation process and are designed to satisfy various business requirements.

Microsoft Smooth Streaming (MSS) [8]: In 2008, Microsoft launched IIS Media Services extension with a new adaptive video streaming over HTTP feature called Smooth Streaming. It was designed to deliver HD videos to viewers. MSS periodically detects network conditions to avoid bandwidth fluctuations. It uses the available bandwidth, playback window resolution, and CPU load at the client side as the metrics for bitrate adaptation. During each streaming session, MSS opens two TCP connections with the server. The first one is used to deliver video segments, while the second one is used for audio, though the two TCP connections could interchange depending on the conditions. MSS showed its efficiency in many sports events like the Beijing Summer Olympic Games 2008, where TV broadcasters used MSS to provide live streaming to 16 million clients [71].

Apple HTTP Live Streaming (HLS) [9], [72]: Due to the popularity of Apple's mobile devices, HLS is the most widely used adaptive video streaming system. Apple Inc. implemented it as part of QuickTime [73] and on iOS devices such as the iPhone and the iPad. It is designed to support both live and on-demand streaming but specifically targets mobile environments. The HLS client makes its bitrate decisions based on network throughput and device capabilities (*e.g.,* CPU, resolution, memory, *etc.*). In an attempt to better utilize the available bandwidth, an HLS client can request many segments at the same time. Furthermore, HLS provides a flexible framework for media encryption. Currently, HLS is natively supported in the Safari Web browser in both iOS and macOS devices, Windows 10 Edge browser [74], and Android 3.0+ devices [75].

Adobe Open Source Media Framework (OSMF) [10]: OSMF is a free, open source software framework for robust adaptive video streaming over HTTP. It was implemented using ActionScript [76] by Adobe systems with the following objectives: *(1)* simplify player development where developers could focus on improving the overall viewer experience, *(2)* offer a set of features for third-party services like rendering, advertising, and reporting, and *(3)* simplify third-party developments by enabling ecosystem partners to focus on delivering best-in-class services instead of player integration. OSMF supports both live and on-demand video streaming, progressive download, sequential and parallel compositions of video, and it adapts to the network variations based on the available bandwidth and device processing capabilities.

The three proprietary streaming solutions described above show efficiency in terms of bitrate adaptation behavior of a single client in response to bandwidth fluctuations. However, several studies [51], [77]–[80] have shown instability issues when multiple clients competed for a bottleneck link in a shared network. From these experiments the following insights were deduced:

- The bitrate adaptation heuristics provide suboptimal bitrate decisions as they fail to adapt quickly to rapid bandwidth variations. Thus, clients suffer from buffer underruns, video instability, quality oscillations, and unnecessary bitrate switches.
- They are not able to ensure a fair viewer experience under some circumstances resulting in low efficiency and poor per-viewer QoE.
- The MSS client outperforms the others, since it achieves the highest playback bitrate, and a low number of bitrate switches during mobile video streaming sessions.
- Based on standard capabilities and features [80], DASH offers nearly everything compared to these proprietary formats.

4) Mixed Adaptation: In this type of scheme, the client makes its bitrate selection based on a combination of metrics including available bandwidth, buffer occupancy, segment size and/or duration.

Other studies have looked at both the available bandwidth and buffer occupancy in order to determine the bitrate of the next segment. Yin *et al.* [81] developed a control-theoretic framework that allows the understanding and exploration of the trade-offs between bandwidth-based and buffer-based adaptation algorithms under different network bandwidth variations. The authors designed a practical model-predictive controller, FastMPC, that optimally combines both bandwidth and buffer size predictions in order to find an appropriate bitrate for the next segment and maximize QoE. A similar approach was also studied in [82]. Li *et al.* [32] formulated the bitrate selection decision as an optimization problem, where at each segment downloading step, the proposed scheme finds an appropriate bitrate that ensures a high and consistent quality subject to bandwidth fluctuations and without risking a buffer depletion. Similarly, Sobhani *et al.* [83] predicted available bandwidth and buffer level using a fuzzy logic mechanism, which is used to select a suitable bitrate. However, these algorithms only ensure a consistent quality at each client without taking the fairness and content type/properties into account when many clients compete for the available bandwidth.

ELASTIC [84] is a fEedback Linearization Adaptive STreamIng Controller, based on feedback control theory [85], that generates a long-lived TCP flow and avoids the ON-OFF steady state behavior which leads to bandwidth overestimations. ELASTIC was introduced to ensure bandwidth fairness between competing clients based on network feedback assistance, but without taking the viewer QoE into consideration. In addition, it ignores quality oscillations in its bitrate decisions. Thus, both during bandwidth fluctuations and in fixed bandwidth environments, ELASTIC may produce a high number of bitrate switches resulting in poor QoE.

Miller *et al.* [86] presented a bitrate adaptation algorithm that uses the current buffer occupancy level, estimated available bandwidth, and average bitrate of the different bitrate levels from the MPD as metrics in its bitrate selection. It aims to *(i)* accurately estimate the available bandwidth and avoid bandwidth overestimation, and *(ii)* maximize the bitrate while minimizing startup delay, number of stalls, quality oscillations, and playback interruptions. The algorithm changes its behavior based on the current buffer level. It can improve the fairness between competing clients, but it does not take any metric of viewer satisfaction into account. Furthermore, in a shared network environment, clients can suffer from video instability, stalls and quality oscillations even when clients reach the highest quality level. This is due to the lack of bitrate decisions which consider viewer QoE.

Jiang *et al.* [87] studied the limitations of video players when a large number of clients shared the same network by providing an experimental study that identified the main factors in bitrate selection. The authors introduced FESTIVE (Fair, Efficient and Stable adapTIVE algorithm), a bitrate adaptation algorithm that aims to improve efficiency, fairness and stability. FESTIVE contains *(a)* a bandwidth estimator module, *(b)* a bitrate selection and update method that tries to avoid unfairness of stateless bitrate selection[2] by making the player stateful, and *(c)* a randomized scheduler that incorporates the buffer size to schedule the download of the next segment. For the same purpose, Throughput-Friendly DASH TFDASH [88] uses a logarithmic-increase-multiplicative-decrease (LIMD) based bandwidth probing algorithm to estimate the available bandwidth and a dual-threshold buffer for the bitrate adaptation.

Tian and Liu [89] offered algorithms that aim to balance bandwidth utilization and smoothness in DASH in both single- and multi-CDN scenarios. Using the buffered video time as a feedback signal, the client is able to adapt the video rate according to the available bandwidth, which is estimated using the support vector regress (SVR) [90] algorithm. Spectrum-based QUality ADaptation (SQUAD) [91] is a lightweight bitrate adaptation algorithm that uses the available bandwidth and buffer information to increase the average bitrate of a video, while minimizing the number of quality switches. For startup, SQUAD follows a conservative approach of fetching more low-quality segments in order to alleviate any inaccuracies in future bandwidth estimations which could result from a single low-quality segment estimation. Later, the algorithm uses the spectrum, which is the variation of the average segment bitrates, and the buffer level to choose the next segment bitrate. Havey *et al.* [92] designed a multi-path solution for rate adaptation in wireless networks. The authors avoided the problems of TCP congestion control by implementing a similar logic at the application layer. Parallel TCP streams have been proven to increase the throughput compared to single TCP streaming. However, this incurs extra request/response overhead and imposes changes on the application stack.

Other studies incorporate more metrics for bitrate selection like the current segment quality, size and download time. SARA [93] is a Segment-Aware Rate Adaptation algorithm that is based on the segment size variation, the available bandwidth estimate, and the buffer occupancy. Since HTTP uses TCP, the throughput of a segment is dependent on the file size, and thus, the authors propose to enhance the typical MPD file to include the size of every segment. For each new segment download, the client decides the new segment quality based on the estimated bandwidth (which is assessed using the segment size) and the current status of the buffer.

ABMA+ [94] is a lightweight adaptation algorithm that selects the highest segment representation based on the estimated probability of video rebuffering. It makes use of buffer maps, which define the playout buffer capacity that is required under certain conditions to satisfy a rebuffering threshold and to avoid heavy online calculations. The authors defined five QoE metrics to evaluate ABMA+ and compared it with BBA and Rate-Based Algorithm (RBA), which are explained in detail in [94]. The authors showed that ABMA+ can efficiently adapt the video representations to the network conditions, while minimizing frequent quality switches. Bentaleb *et al.* [95] discussed the shortcomings of the existing client-based schemes. To sidestep these drawbacks, the authors leveraged a game theory [96] framework and developed the GTA (Game Theory Adaptive bitrate) scheme. GTA uses a cooperative game in coalition formand then formulates the bitrate selection problem as a bargaining process and consensus mechanism. Thus, the DASH clients can create an agreement among themselves and achieve their QoE objectives. GTA improves the viewer QoE and video stability without increasing the stall rate or startup delay.

5) MDP-Based Adaptation: In Markov Decision Process (MDP)-based adaptation, the video streaming process is formulated as a finite MDP to be able to make adaptation decisions under fluctuating network conditions. Xing *et al.* [97] proposed a real-time best-action search algorithm over multiple access networks that aims to produce smooth and high-quality video playbacks. The authors used both Bluetooth and WiFi links to simultaneously download video segments. In each state, the MDP was formulated so the rate adaptation agent takes the buffer level, SVC layer index, Bluetooth traffic, available bandwidth, and the index of each segment fetchedas inputs. The reward function is designed to consider

[2]Stateless bitrate selection refers to selecting the highest bitrate lower than the available bandwidth.

the average playback quality, interruption rate, and playback smoothness. However, this scheme shows limitations during user mobility which negatively affect the viewer QoE. The mobility problem was addressed by Bokani *et al.* [98] who modeled the bitrate adaptation logic as an MDP problem in vehicular environments. A three-variant of Reinforcement Learning (RL)-based algorithms were introduced. These algorithms take advantage of the historical bandwidth samples to build an accurate bandwidth estimation model.

Another noteworthy work is Petrangeli *et al.* [99]. The authors tackled the problem of QoE and fairness when multiple clients compete at a bottleneck link and they proposed a multi-agent RL-based bitrate adaptation scheme that uses a central manager in charge of collecting QoE statistics (segment bitrate) and coordination between the competing clients. The developed algorithm ensures a fair QoE distribution among the competing clients and improves viewer QoE, while avoiding suboptimal decisions. However, this model does not consider stalls and quality switches which can lead to rebuffering events. Unlike [99], Chiariotti *et al.* [100] developed an MDP-based online bitrate adaption algorithm for DASH clients that aimed to select the optimal representation, maximizing the long-term expected reward (QoE). This reward function was calculated from a combination of quality oscillations, segment quality, and stalls experienced by the client. The authors used RL to gather information on the network environment through experience to approach an optimal solution. To avoid slow convergence and suboptimal solutions caused during the RL process, the authors exploited a parallel learning technique. Zhou *et al.* [101] tackled a similar problem by proposing mDASH to improve viewer satisfaction during long-term bandwidth variations. The authors first formulated the bitrate adaptation logic as an MDP optimization problem where the buffer size, bandwidth conditions, and bitrate stability were taken as Markov state variables. They subsequently solved this problem by proposing a low-complexity greedy suboptimal algorithm. Compared to previous MDP-based studies, Pensieve [102] and Deep Q-Learning DASH (D-DASH) [103] were proposed to improve accuracy and speed of bitrate decision estimations using Deep Reinforcement Learning (Deep RL) [104]. Pensieve [102][3] is a framework that is built based on observations collected by DASH clients (*i.e.,* throughput estimation and buffer occupancy) across large video streaming experiments. It does not rely on pre-programmed models or assumptions about the environment, but, in fact, gradually learns the best policy for bitrate decisions through observation and experience. D-DASH [103] combines deep learning and reinforcement learning mechanisms to improve the QoE for DASH, and achieves a good trade-off between policy optimality and convergence speed during the decision process. In particular, it uses mixed learning architectures including feed-forward and recurrent deep neural networks with advanced strategies. Both solutions [102], [103] perform adequately and present the benefits of incorporating Deep RL with ABR heuristics in the bitrate decision process. The proposed MDP-based schemes yield a significant improvement in the overall performance in terms of stalls, and thus, ensure an acceptable level of viewer QoE. However, they may suffer from instability, unfairness, and underutilization when the number of clients increases, probably because such factors are not taken into account in the MDP models and due to clients' decentralized ON-OFF patterns.

[3]A pensieve is a device used in Harry Potter [105] to review memories.

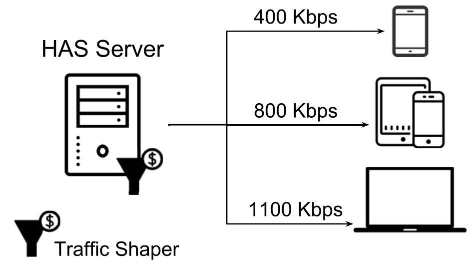


Fig. 10. Server-based bitrate adaptation.

B. Server-Based Adaptation

Server-based schemes use a bitrate shaping method at the server side and do not require any cooperation from the client (see Fig. 10). Thus, the switching between the bitrates is implicitly controlled by the bitrate shaper. The client still makes its own decisions, but the decisions are more or less determined by the shaping method on the server.

Traffic shaping methods have been deployed in [106] and [107] where the authors analyzed instability and unfairness issues in the presence of multiple HAS players competing for the available bandwidth. These studies proposed a traffic shaping method that can be deployed at a home gateway to improve fairness, stability and convergence delay [107], and to eliminate the OFF periods during the steady states (the root cause of the instability problem) [106].

To improve the live experience, Detti *et al.* [108] proposed a tracker-assisted adaptation strategy in the presence of network caches. The proposed architecture consists of clients communicating with a server through a shared proxy and a server having a tracker functionality that manages the clients' statuses and helps them share knowledge about their statuses. De Cicco *et al.* [109] proposed a feedback control theory-based algorithm called Quality Adaptation Controller (QAC). QAC aims to control the size of the server sending buffer in order to adjust and select the most appropriate bitrate level for each DASH player. It aims to maintain the playback buffer occupancy of each player as stable as possible and to match bitrate level decisions with the available bandwidth. Bruneau-Queyreix *et al.* [110] developed the MS-Stream system, a multiple-source adaptive streaming solution to improve viewer QoE, where the client fetches the segments (divided into a set of subsegments and stored in the servers) from multiple MS-Stream servers.

The server-based bitrate adaptation schemes produce high overhead on the server side with a high complexity,[4] especially when the number of clients increases. These schemes

[4]The server needs to store and maintain the information for each client to perform bitrate adaptation.

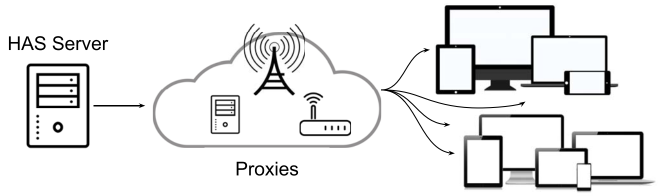


Fig. 11. Network-assisted bitrate adaptation.

also need modifications to the MPD [108] or a custom server software to implement the bitrate adaptation logic [106]–[109]. This may be perceived as a violation of the DASH standard design principles, namely that the server should be a standard HTTP server, and that the bitrate adaptation algorithm should, consequently, run at the client side. The server and network-assistance approach [16], [111] can be an alternative solution, where in-network entities and servers aid the client in its bitrate decisions. This approach is discussed in detail in Section III-D2.

C. Network-Assisted Adaptation

The network-assisted approach depicted in Fig. 11 allows the HAS clients to take in-network decisions during the bitrate adaptation process into consideration. This happens by collecting measurements about the network conditionswhile informing the clients on the suitable bitrates to be selected. The in-network process needs a special component (*e.g.,* agent/proxy deployed in the network) to monitor the network status and conditions. It offers network-level information that allows the HAS clients to efficiently use network resources.

QoE-aware DASH (QDASH) [112] is a proxy between the clients and the streaming server that aims to avoid video oscillations by ensuring a gradual change in bitrate levels using integrated intermediate levels, which can lead to a better QoE. QDASH consists of a QDASH-abw module to measure the bandwidth and a QDASH-qoe module that assists the client in choosing a suitable bitrate that can support the current network conditions and buffer occupancy. However, it generates significant overhead in the network, especially with increasing client numbers. This overhead may eventually lead to network congestion in itself, resulting in a low QoE. Similarly, Bouten *et al.* [113] tackled the problem of multiple DASH clients competing for the available bandwidth by proposing a QoE-driven in-network optimization system for adaptive video streaming. The proposed system consists of a set of agents deployed along the path between the clients and streaming server, where they play the role of proxies. These network agents periodically measure and monitor the available bandwidth along the path using packet sampling techniques and solve an optimization problem to determine the optimal bitrate for the next segments to be downloaded. This information is then sent to the clients. However, similar to [112], it can generate significant overhead and is not resilient to agent failures. To reduce buffer underrun events and improve the client's viewing experience, Krishnamoorthi *et al.* [114] presented BUFFEST, a classification framework for real-time prediction of the client's buffer conditions from both HTTP and HTTPS traffic. It consists of an event-based buffer emulator module and an automated training online classifier that are responsible for accurately tracking/predicting the client's buffer conditions and TCP/IP packet-level traffic classification, respectively. For the same aim and inspired by the Network Utility Maximization (NUM) [115] framework, D'Aronco *et al.* [116] proposed a distributed price-based, network-assisted HAS system for multiple concurrent HAS clients sharing a common bottleneck. The proposed solution introduces the definition of a price (*i.e.,* a function of the segment download times that are captured by the HAS clients), which is inspired by a congestion control algorithm. Then, using the price information, a central coordinator assists the clients in their decisions to maximize overall user satisfaction and QoE fairness.

To alleviate overhead-caused network performance degradation, Petrangeli *et al.* [117] tried to avoid fairness issues when multiple HAS clients consume video at the same time and compete for shared network resources by proposing a QoE-driven in-network bitrate adaptation algorithm named FINEAS (Fair In-Network Enhanced Adaptive Streaming). To achieve fairness, FINEAS uses in-network components such as proxies that offer information about network conditions like currently available bandwidthand suggestions about the best bitrate. Each client may use these suggestions as a criterion for bitrate selection. FINEAS shows good performance in homogeneous systems but in the real world, heterogeneous devices with different characteristics exist. Thus, sharing the bandwidth equally among competing clients may result in high QoE on some devices but low QoE on others. In [118], Network Optimization for Video Adaptation (NOVA) was proposed to fairly maximize viewer QoE while avoiding unnecessary bitrate switching in a heterogeneous environment. The authors formulated the multi-client competition issue as an optimization problem subject to buffer occupancy, network conditions and delivery cost. Thereafter, NOVA tries to find the optimal bitrate for each client. NOVA consists of two main elements: bandwidth allocation and quality adaptation. While NOVA achieves good QoE compared to traditional DASH systems, the efficiency of the proposed architecture relies on strong statistical assumptions such as stationary ergodicity, which may negatively impact the convergence time during the search for optimal decisions [119].

Many studies [120]–[127] have proposed bitrate adaptation schemes to improve viewer QoE in cellular networks. AVIS [120] is a network-based radio resource allocation framework designed for adaptive video flows in cellular networks. It can optimally allocate resources for each client (separating DASH flows from others) and ensure fairness and stability between them while maintaining high resource utilization. Similarly, Kleinrouweler *et al.* [122] installed HTTP proxies at the network gateways that evenly allocated the available bandwidth between the streaming clients. The proxy re-writes client requests that demand a bitrate higher than the one designated by the proxy, and also adds an HTTP header to the response informing the client of the change. The streaming process was modeled as an MDP, where each state represents the number of active clients and the transitions between the states are linked to starting and stopping the players. To account for stability, the number of switches relates to the frequency of transitions between the MDP states. In contrast, El Essaili *et al.* [121] developed a QoE optimizer and resource manager framework that can dynamically find the optimal bitrate for a subject to wireless channel conditions, buffer levels, and achievable QoE. It allocates the required bandwidth for each client based on its QoE unlike [120], [122] where all clients receive an equal share of the allocated bandwidth, which does not necessarily mean that all the clients enjoy a good experience due to intrinsic differences across the device capabilities.

In the same context, the Rebuffering Aware Gradient Algorithm (RAGA) [123] is a cross-layer buffer-aware wireless resource allocation algorithm that considers only the playback buffer size during the bitrate selection process. It makes use of DASH's standardized user feedback from the buffer, both its level and rate of level changes. The same authors later proposed a new architecture to enhance the QoE in LTE networks [124]. The architecture consists of a Video Aware Controller (VAC) at the network core that acts as a central intelligence unit for translating the video qualities and buffer levels into QoS parameters. The authors also proposed a new algorithm that computes the dynamic Maximum Bit Rate (dynamic-MBR) for each client based on its buffer level obtained from the feedback. Han *et al.* [125] proposed Multi-path DASH (MP-DASH), a multi-path framework with awareness of the network interface preferences of the clients. It aims to improve multi-path TCP (MPTCP) to support DASH considering the user network interface preferences, thus enhancing the efficiency of video delivery without sacrificing viewer QoE. MP-DASH consists of two main components including the MP-DASH scheduler and the video adapter. The scheduler takes user interface preferences, segment size and its delivery time from the DASH client into consideration. Based on this, it decides the best way to fetch the segment over multiple paths. The video adapter is a lightweight add-on to existing client-based adaptation schemes to be multi-path friendlier, being responsible for handling the interaction between the bitrate adaptation scheme, and the MP-DASH scheduler.

To reduce video instability, QoE unfairness and stalls in cellular networks, Yan *et al.* [126] designed Prius as a framework that consists of a hybrid edge cloud and a client-based adaptation scheme. Similarly, Zahran *et al.* [127] proposed a Stall-aware Pacing (SAP) traffic management solution for DASH clients. It aims to reduce video stalls while maintaining a consistent QoE when multiple DASH clients with diverse channel conditions compete for resources. SAP leverages both network and client state information to optimize the per-player QoE. Leveraging Machine Learning (ML) [128] mechanisms, De Grazia *et al.* [129] developed a multi-stage ML cognitive approach for DASH when multiple clients compete for the available bandwidth in a shared channel. The proposed solution incorporates unsupervised and supervised ML to comprehend the Quality-Rate (Q-R) relationship. The authors deployed a cognitive HTTP proxy (CHP) that was responsible for controlling the video traffic towards the clients, performing traffic classification, helping clients in their decisions, and applying resource allocation according to the Q-R function. Motivated by the fact that TCP connections are well-modeled as traversing a piecewise-stationary sequence of network states [130], Akhtar *et al.* [131] designed Oboe which allows the automatic tuning of configuration parameters to different network conditions for an ABR scheme. Consequently, these configuration parameters are applied at run-time to match the current network state. The proposed system significantly improves the bitrate decision of client-based adaptation schemes like BOLA [67] and FastMPC [81], and it offers a 24% on average better viewer QoE compared to Pensieve [102].

Other approaches incorporate OpenFlow-enabled solutions with HAS. Georgopoulos *et al.* [132] proposed an OpenFlow-based in-network caching service, named OpenCache, that leverages software defined networking (SDN) to optimize video-on-demand DASH streams. OpenCache uses SDN to provide cache-as-a-service (CaaS) for media content and aims to alleviate last mile scalability issues by pushing the DASH segments as close to the client as possible without requiring any modifications in the delivery method, and to improve network resource utilization and QoE for the viewers. Additionally, it can provide network and DASH clients' measurements that help CDN providers to enhance content placement and delivery mechanisms. Cofano *et al.* [133] investigated video quality fairness (VQF) for cases in which multiple heterogeneous adaptive streaming players share the same bottleneck link. The authors proposed a Video Control Plane (VCP) that enforces a video quality management policy to ensure fairness. VCP was implemented on top of an SDN controller as a network controller application and consists of three network-assisted streaming approaches: bandwidth reservation, bitrate guidance and hybrid between bitrate guidance, and bandwidth reservation. Bhat *et al.* [134] designed an SDN-assisted architecture for HAS systems, termed SABR. This method leverages SDN capabilities to assist and manage HAS players and it collects various information such as available bandwidth and client states to guide player bitrate decisions. Seema *et al.* [135] developed a DASH-based video platform for miniaturized devices including sensors, called Wireless Video Sensor Node Platform DASH (WVSNP-DASH). The proposed platform uses an alternative approach to segment the video to be convenient for miniaturized wireless devices and sensors. It utilizes a specific naming syntax (based on a simplified Backus-Naur Form [136]) for video segments such that each segment is an independently playable file that embeds essential metadata required for video playout in its name. In this way, the client can play the segment without requiring to download the manifest file and initial segments. WVSNP-DASH is designed based on core elements of HTML5 (*e.g.,* HTML5 File System). Also, it can encapsulate any container, codec and DRM that are supported by a Web browser. However, this paper does not analyze the overhead introduced by WVSNP-DASH, *i.e.,* the new data embedded in each segment which may significantly impact the network efficiency and lifetime. For bitrate adaptation schemes over Information-Centric Networking (ICN), Lederer *et al.* [137] investigated the possibilities of integrating HAS over ICN. The authors highlighted use cases and scenarios, namely Netflix-like video streaming, peer-to-peer (P2P) uses, video sharing, and IPTV. Additionally, the authors presented available tools and testbeds to evaluate HAS over ICN, and highlighted several challenges and open issues. Further details of the HAS over ICN architecture can be found in RFC 7933 [138]. The performance of DASH over ICN is examined by Rainer *et al.* [139]. The authors analyzed the performance gap between different ICN-based forwarding strategies with their theoretical optimum at the network level and various client-based adaptation schemes at the application level. They derived the theoretical optimum bound by formulating the concurrent streaming clients in ICN as a fractional Multi-Commodity Flow Problem (MCFP) with and without caching, showing that HAS performance can be improved by benefiting from ICN multi-path and caching capabilities. Petrangeli *et al.* [140] focused on combining HAS and SVC over ICN networks. They used SVC mainly for the following reasons: *(i)* SVC allows to fully exploit the benefits of ICN while avoiding suboptimal bitrate selections, *(ii)* it helps the clients to mitigate bandwidth overestimation, and *(iii)* the layered structure of SVC enables the benefits from ICN's multi-path capabilities Xu *et al.* [141] proposed EcoMD, an ICN-based cost-efficient multimedia content delivery solution for vehicular ad hoc networks (VANETs) to reduce the cost of video delivery in highly dynamic VANETs. The authors first analyzed two essential factors, namely content mobility and supply-demand balance. Then they formulated the cost associated with video delivery as a Mixed Integer Programming (MIP) optimization problem. Finally, they proposed three adaptive heuristic solutions to solve the optimization problem: *(1)* priority-based path selection, *(2)* least-required sources maintaining, and *(3)* on-demand in-path caching enhancement. Similarly, Detti *et al.* [142] proposed an ICN-based P2P streaming application for live HAS systems over cellular networks. The main insight of this work is to show the possibility of exploiting ICN capabilities to provide a good HAS service and achieve a simplified deployment process. In the application, the HAS clients (or peers) construct a P2P one-hop mesh network that enables cooperative downloading of the same live video. These clients use their cellular network interfaces to connect to the HAS server and are connected to each other through proximity WiFi channels.

In general, the presented ICN-based solutions use heuristic information (collected from the requested content) to perform the caching decision by a special node. Some of these solutions produce a large number of redundant copies, and thus, impact storage resources. Providing efficient content management, ensuring high cache performance, and designing a robust HAS delivery system over ICN are still open issues.

D. Hybrid Adaptation

In hybrid bitrate adaptation, many networking entities collaborate together and collect useful information about network conditions that can help HAS clients in their bitrate selection. This type of technique consists of SDN-based and server-and-network-assisted adaptations.

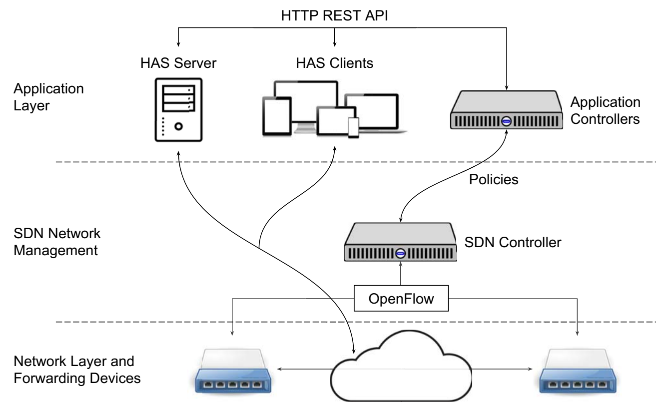


Fig. 12. Architecture for SDN-based bitrate adaptation.

1) SDN-Based Adaptation: Two key insights of integrating SDN [143], [144] within an adaptive video streaming system are as follows:

- SDN allows for network resource control and monitoring capabilitiesand thus simplifying network resource programming and deployment.
- Pure client-driven bitrate adaptation algorithms show their limitations when a set of DASH clients compete in a shared network environment and when the network size grows, resulting in issues such as video instability, quality oscillations, buffer underruns, unfairness, and underutilization. These issues are largely due to a lack of coordination among the clients, which could be ensured by a central mechanism that has the global network view in a manageable network environment (*e.g.,* a last mile like campus network). With a central coordinator and the integration of such coordination information, these issues can be avoided and viewer satisfaction can be improved.

Fig. 12 depicts SDN-based bitrate adaptation, where the network resources and competing clients are controlled and monitored by a central component in the control plane, more precisely the SDN controller.

Georgopoulos *et al.* [145] proposed an SDN-assisted QoE Fairness Framework (QFF), which sought to optimize QoE by ensuring video quality fairness among multiple competing DASH clients in the last mile. The proposed framework leverages OpenFlow to monitor the quality of the video streams and allocate/manage resources in the network.

The same authors later proposed an improvement of QFF by introducing the SDN-based in-network QoE measurement framework IQMF [146], which acts as a proxy and aims to provide per-client transparent monitoring of QoE during the video session, and subsequently offers its feedback to network and content providers through a well defined API. IQMF was proposed due to the fact that traditional network-level metrics like bandwidth, packet loss, jitter, and end-to-end delay could not provide an estimation of video quality. Both, QFF and IQMF take only two metrics into account, device resolution and available bandwidth, without considering the buffer level. Thus, clients may be subject to buffer starvation.

Nam *et al.* [147] proposed an SDN-based application that aims to manage network resources while monitoring network conditions and client feedback (QoE metrics), when multiple clients compete for a shared capacity. The SDN application dynamically reroutes the video flows using the Multiprotocol Label Switching (MPLS) traffic engineering mechanism over SDN when QoE requirements are violated (during buffer underrun events, for instance). Such an approach can improve the overall QoE by selecting the best path to the server. However, the authors do not describe the time effect of dynamic path changes during the streaming session. This problem was addressed by Wang *et al.* [148] through the development of GENI Cinema (GC), an SDN-assisted service for live video streaming. GC aims to provide online live educational video streaming among many campuses using the GENI SDN-based network resource infrastructure. Steaming clients can upload and/or watch online videos via a public shared Web portal, and the GC service is able to monitor and manage the video flows and resources over one or multiple routes dynamically using SDN features. The GC service has been shown to provide scalable, stable, and fair live video streaming.

Petrangeli *et al.* [149] proposed an SDN-based framework that aimed to reduce video freezes caused by sudden bandwidth fluctuation by applying a prioritization technique during the segment delivery process. The SDN controller represents the main component of the proposed framework, where it is responsible to collect the network status information such as bandwidth changes, latency, and statuses of the HAS clients. Based on this information, the controller decides whether a segment has to be prioritized or not in order to alleviate video freezing at the client. In the same context, Kleinrouweler *et al.* [150] described an SDN-based network architecture for DASH that aims to ensure stable and high quality video delivery, while avoiding the mismatch between the TCP mechanism and the dynamic bursty nature of DASH traffic. The proposed architecture consists of three layers: SDN network application controllers, SDN network management, and programmable network infrastructure. The SDN network application helps the set of competing DASH clients in their bitrate selection, while the SDN network management uses a dynamic queue-based mechanism for QoS provisioning. However, the proposed architecture does not consider device heterogeneity, which is important for determining the fair share of available bandwidth and QoE fairness. To address these issues, Bentaleb *et al.* [151] proposed a new end-to-end SDN-based resource allocation and management architecture for HAScalled SDNDASH. The proposed architecture leverages SDN capabilities to manage and allocate network resources for each client based on its QoE. It consists of the three layers application, control, and network, as well as six core entities within those layers: DASH server, DASH clients, SDN-based external application, SDN controller, SDN-based internal application, and forwarding devices. SDNDASH formulates the QoE maximization and optimal decision for both bitrate and network resource allocation as a maximization optimization problem, leading to significant improvements in per-client QoE while avoiding HAS stability issues. For the same context, SDNHAS [152] and ORL-SDN [153] were developed. The former was proposed to resolve three limitations that were not addressed in SDNDASH and could affect the ABR decisions, namely: *(i)* the difficulty to support large-scale deployments of HAS players, *(ii)* non-trivial communication overhead, and *(iii)* limited support for system heterogeneity. The latter is an online reinforcement learning (RL) QoE optimization framework for SDN-enabled HAS systems. The proposed framework consists of three phases. First, it groups the HAS players into a set of disjoint clusters based on a perceptual quality index. Second, it formulates the bitrate selection as a Partially Observable Markov Decision Process. Third, it implements an online Q-learning algorithm to solve the QoE optimization problem and find in parallel the optimal bitrate decision for each cluster.

To improve the viewer QoE in the context of HAS in hybrid fiber coax (HFC) network environments, an SDN-based bandwidth broker solution [154] termed BMS (Bandwidth Management Solution) was developed. BMS formulates the bitrate decisions as a convex optimization problem, which relies on a concave network utility maximization (NUM) function. BMS is proposed to meet per-session and per-group QoE objectives. Thus, BMS is able to avoid common HAS issues like video instability, unfair and unequal quality distribution, and network resource underutilization.

Lai *et al.* [155] proposed an SDN-based manager in 5G OpenFlow-enabled wireless networks for HLS services. The manager aims to allocate a suitable on-demand network resource (*e.g.,* bandwidth) that improves the QoE taking into consideration the media segment perceptual quality and client buffer size during bitrate selection. However, the authors consider neither the radio characteristics that exhibit sudden bandwidth fluctuations nor the handover situations due to user mobility.

All theses studies as well as C3 [156], CFA [157], CS2P [158], Pytheas [159] share a common characteristic, which is that there exists a central controller to control, manage and monitor HAS traffic. However, these solutions do not scale well and support system heterogeneity. They also generate additional overhead that can affect the network performance.

2) Server and Network-Assisted Adaptation: Thomas *et al.* [16], [111], [160] were motivated by the fact that the client-driven approach of DASH left less control to the network and service providers, which introduced new challenges for them in service differentiation, and proposed the Server and Network-assisted DASH (SAND) architecture. SAND is a control plane that offers asynchronous client-to-network, network-to-client, and network-to-network communications. SAND allows to collect metrics and status information from different entities in the system including the clients and to send feedback to the clients and DASH-aware Network Elements (DANE) including the servers, caches and other network entities along the media path. This feedback is used by the clients to assist in the bitrate adaptation and by the DANEs to improve media delivery. To enable the communication between the clients and DANEs, SAND defines the following interfaces to carry various types of messages:

- Client-to-Metrics-Server and Client-to-DANE Interfaces carry the metrics and status messages, respectively.

- DANE-to-DANE Interface carries the parameters enhancing delivery messages.
- DANE-to-Client Interface carries the parameters enhancing reception messages.

The SAND architecture is primarily based on feedback from the clients (*e.g.,* QoE metrics) and the network (*e.g.,* available bandwidth). This kind of architecture is not easy to implement, and hence, only few works have tackled this problem yet. Unsurprisingly, SDN is one of the main enablers for the SAND architecture [151], [161]. Further details on the SAND architecture and messages can be found in ISO/IEC 23009-5, which was published by MPEG in early 2017.

IV. Comparison Between Bitrate Adaptation Schemes

Each adaptation scheme proposes distinct criteria for bitrate decisions, where they work only under indirect or implicit assumptions and specific scenarios, and focuses on a specific deployment or different network characteristics. Currently, there is a lack of general consistent frameworks that can formally evaluate and compare different bitrate adaptation schemes, and test and verify the efficiency of their components. Only a few algorithms formally describe what objective they want to optimize, making an effective comparison nigh impossible. In this part, we provide a feature comparison between various state-of-the-art bitrate adaptation schemes in each category from the taxonomy in Fig. 3. Table II summarizes this comparison for each surveyed paper in terms of the following aspects:

- *Heuristic(s):* The measurements and values that the algorithm bases its download decision on {*BW:* available bandwidth, *Buffer:* buffer occupancy, *SDT:* segment download time, *DC:* device capabilities, *CPU:* CPU load, *QT:* perceptual quality, *PA:* proxy assistance, *CA:* central entity assistance, *SDN* or *SDN-app:* SDN assistance, *Seg-size:* segment size, *Seg-quality:* segment quality, *Seg-schedule:* segment scheduling}.
- *Fairness:* Describes the algorithm's fairness between multiple clients that share the network. Some algorithms equally share the bandwidth among the clients, indicated by BW, others share the bandwidth based on either perceptual quality or QoE, indicated by QT and QoE, respectively.
- *QoE:* Does the algorithm support and integrate one of the objective QoE models?
- *Number of clients:* Single indicates one client only, multiple(few) indicates less than 10 clients, and multiple(many) indicates more than 10 clients.
- *QoE optimization:* Does the algorithm propose a QoE model and aim to optimize it?
- *Content type:* Live or video-on-demand (VoD).
- *Heterogeneity:* Does the algorithm take heterogeneous devices into account in its experimental testing?
- *SVC support:* Does the adaptation algorithm support the streaming of SVC-encoded video?
- *BG traffic:* Does the paper include background traffic in their experimental tests?

From Table II, we can deduce the following outcomes:

- The client-based adaptation schemes show a good performance given certain environments and circumstances. They are suited for large-scale deploymentsand they require modifications only on the client side. However, most of these schemes suffer under network-bottleneck conditions (*i.e.,* they are not globally optimal). Reason for this is the lack of a central element that guides the players in their bitrate decisions.
- The server-based adaptation schemes provide the advantage of central control. However, these schemes may introduce a high complexity on the server and produce additional overhead, which may harm the network efficiency. Additionally, these schemes need modifications in the manifests and/or the server side.
- The network-assisted adaptation schemes aim to have a general view of the network, which helps the clients in their bitrate decisions. These schemes are suitable for small-to-large networks and show a high performance in improving the viewer QoE. A similar observation can be made for hybrid schemes. However, the real-world deployment of both scheme classes remains challenging as they introduce some overhead that may harm the network performance and since they require additional entities in the network.

It might be of interest to note that Table II provides only a feature comparison between the different schemes, such as the heuristics, experimentation parameters and collected metrics. A performance comparison is difficult for mainly three reasons: *(i)* the unavailability of source codes, *(ii)* the lack of a unified QoE framework and metrics to evaluate these schemes, and *(iii)* because every scheme has its own parameters and assumptions, and may have been designed for a specific environment and settings.

V. Discussion

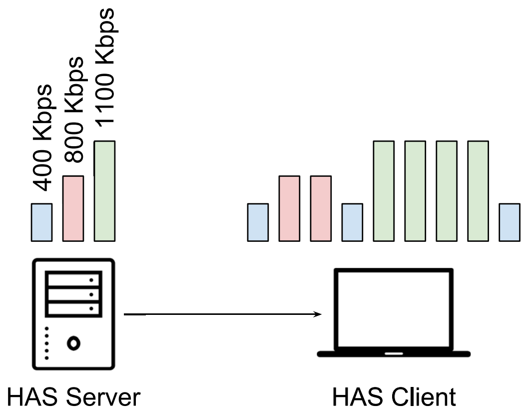
In this section, we discuss emerging HAS trends, namely *(A)* HAS and scalable video coding (SVC), *(B)* advanced transport options such as HTTP/2 and Quick UDP Internet Connections (QUIC), *(C)* immersive media streaming, specifically 360-degree video streaming, and *(D)* HAS datasets.

A. HAS and Scalable Video Coding (SVC)

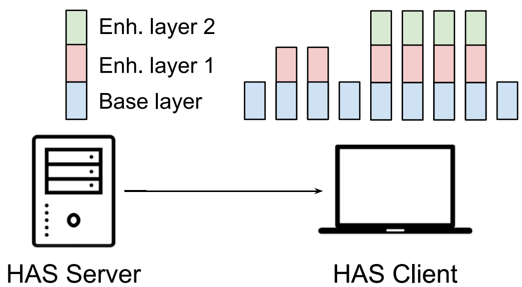
In most state-of-the-art adaptive streaming systems, non-scalable video coding (*i.e.,* AVC, HEVC, VP9/AV1) is widely relied upon due to its coding efficiency, ease of implementation, and widespread adoption. However, scalable video coding has multiple benefits such as resiliency to packet losses and better adaptability to device capabilities (*e.g.,* if a device is not capable of decoding high-quality videos, it can choose to decode lower layers only). Many studies [55], [68], [162], [163] have shown benefits of using SVC in HAS rather than AVC [30]: *(1)* it allows HAS to support heterogeneous clients, *(2)* it reduces storage and networking costs, and *(3)* it enables CDNs and caches to be used more efficiently, *e.g.,* by prioritizing the base layer and providing enhancement layers only when network resources are available. Fig. 13 depicts SVC-based HAS where each segment can

TABLE II
FEATURE COMPARISON BETWEEN THE SURVEYED BITRATE ADAPTATION SCHEMES

	Adaptation Scheme	Heuristic(s)	Fairness	QoE	Number of Clients	QoE Optimization	Content Type	Heterogeneity	SVC Support	BG Traffic
III-A	[51]	BW	✗	✗	Single	✗	VoD	✗	✗	✗
	[52]	BW	✓(BW)	✗	Multiple(few)	✗	VoD	✗	✗	✓
	[53]	BW	✓(BW)	✗	Multiple(few)	✗	VoD/Live	✗	✗	✓
	[54]	BW	✓(BW)	✓	Multiple(few)	✗	VoD	✗	✗	✓
	[108]	BW	✓(BW)	✗	Multiple(few)	✗	Live	✓	✗	✓
	[58]	BW	✗	✓	Single	✗	VoD/Live	✗	✗	✗
	[98]	BW	✗	✓	Single	✗	VoD	✗	✗	✓
	[55]	BW	✗	✓	Single	✗	VoD	✗	✓	✗
	[56]	BW	✗	✓	Single	✓	VoD	✗	✗	✗
	[64]	BW	✗	✓	Single	✗	VoD	✗	✗	✓
	[65]	Buffer	✗	✓	Multiple(few)	✗	VoD/Live	✗	✗	✗
	[66]	Buffer	✗	✗	Multiple(few)	✗	VoD	✗	✗	✗
	[67]	Buffer	✗	✓	Single	✓	VoD	✓	✗	✓
	[68]	Buffer	✗	✓	Single	✗	VoD	✓	✓	✗
	[8], [9]	BW, DC	✗	✓	Multiple(few)	✗	VoD/Live	✓	✗	✗
	[10]	BW, CPU	✗	✓	Multiple(few)	✗	VoD/Live	✓	✗	✗
	[81], [102]	BW, Buffer	✗	✓	Single	✓	VoD	✗	✗	✗
	[32]	BW, Buffer, QT	✓(QT)	✓	Multiple(few)	✓	VoD/Live	✗	✗	✓
	[82], [84]	BW, Buffer	✓(BW)	✓	Multiple(few)	✗	VoD	✗	✗	✓
	[86]	BW, Buffer	✗	✗	Single	✗	VoD	✗	✗	✓
	[87]	Buffer, Seg-schedule	✓(BW)	✓	Multiple(few)	✗	VoD	✗	✗	✗
	[88]	BW, Buffer	✓(BW)	✓	Multiple(few)	✗	VoD	✗	✗	✗
	[93]	BW, Buffer, Seg-size	✗	✓	Single	✗	VoD	✗	✗	✗
	[89]	BW, Buffer	✗	✓	Single	✗	VoD	✗	✗	✓
	[91]	BW, Buffer	✗	✓	Single	✗	VoD/Live	✗	✗	✓
	[94]	SDT, Buffer	✓(BW)	✓	Multiple(few)	✗	VoD	✗	✗	✓
	[99]	BW, Buffer	✓(QoE)	✓	Multiple(few)	✓	VoD	✗	✗	✓
	[101]	BW, Buffer	✗	✓	Single	✓	VoD	✗	✗	✓
	[92]	BW, Buffer	✓(BW)	✗	Multiple(few)	✗	VoD	✓	✗	✓
	[100]	BW, Buffer, Seg-quality	✗	✓	Single	✓	VoD/Live	✗	✗	✗
	[97]	BW, Buffer	✗	✓	Single	✗	VoD	✗	✓	✓
	[83]	BW, Buffer	✓(BW)	✓	Multiple(few)	✓	VoD	✗	✗	✓
	[69]	Buffer	✓(BW)	✓	Multiple(few)	✓	VoD	✗	✗	✓
III-B	[106], [107]	BW, Server traffic shaper	✓(BW)	✗	Multiple(many)	✗	VoD/Live	✗	✗	✗
	[109]	BW, Buffer	✓(BW)	✗	Multiple(many)	✗	Live	✗	✗	✓
	[110]	BW, Multiple servers	✓(BW)	✓	Multiple(many)	✗	VoD/Live	✓	✗	✓
III-C	[112]	BW, Buffer, PA	✗	✓	Multiple(few)	✓	VoD	✗	✗	✓
	[113]	BW, Buffer, PA	✗	✓	Multiple(many)	✓	VoD	✗	✗	✓
	[117]	BW, Buffer, PA	✓(BW)	✓	Multiple(many)	✓	VoD	✗	✗	✓
	[118]	BW, Buffer, PA	✓(QoE)	✓	Multiple(few)	✓	VoD	✗	✗	✗
	[116], [121]	BW, Buffer, PA	✗	✓	Multiple(few)	✓	VoD/Live	✓	✗	✗
	[133]	BW, Buffer, SDN	✓(QoE)	✓	Multiple(few)	✗	VoD	✓	✓	✗
	[123]	Buffer	✓(QT)	✓	Multiple(few)	✗	VoD	✓	✗	✓
	[124]	CA	✓(QT)	✓	Multiple(many)	✗	VoD	✗	✗	✗
	[120]	BW	✓(BW)	✓	Multiple(few)	✗	VoD	✓	✗	✓
	[122]	BW, PA	✓(BW)	✓	Multiple(many)	✗	VoD	✓	✗	✗
	[132]	BW	✓(QoE)	✓	Multiple(few)	✗	VoD	✗	✗	✓
	[134]	BW, SDN	✓(BW)	✓	Multiple(many)	✓	VoD/Live	✓	✗	✓
	[126], [127]	BW, CA	✓(QoE)	✓	Multiple(few)	✓	VoD/Live	✓	✗	✓
	[125]	BW, Seg-size, Seg-schedule, CA	✓(BW)	✓	Multiple(few)	✓	VoD/Live	✓	✗	✓
III-D	[145]	BW, Buffer, SDN-app	✓(QoE)	✓	Multiple(few)	✓	VoD/Live	✓	✗	✗
	[146]	BW, Buffer, SDN-app	✓(QoE)	✓	Multiple(few)	✗	VoD/Live	✓	✗	✗
	[147]	BW, Buffer, SDN-app	✗	✓	Multiple(many)	✓	VoD	✗	✗	✓
	[148]	BW, SDN	✗	✗	Multiple(many)	✗	VoD/Live	✓	✗	✗
	[151], [152], [154]	Hybrid, SDN	✓(QoE)	✓	Multiple(many)	✓	VoD/Live	✓	✗	✓
	[150]	BW, DC, SDN	✓(BW)	✓	Multiple(few)	✗	VoD	✗	✗	✓
	[149]	Buffer, QoE, SDN	✓(BW)	✓	Multiple(few)	✗	VoD	✓	✗	✓
	[155]	BW, Buffer, SDN	✗	✗	Single	✗	Live	✗	✗	✗

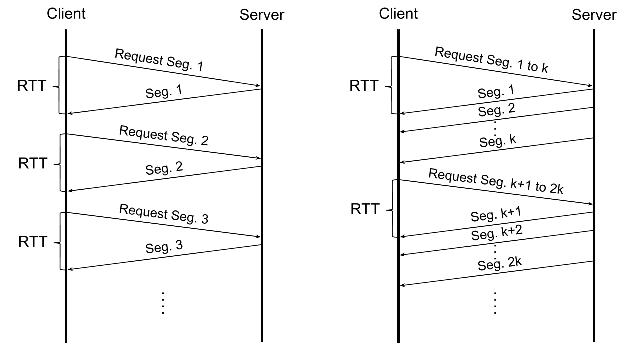


(a) AVC-based HAS.



(b) SVC-based HAS.

Fig. 13. AVC-based vs. SVC-based HAS.



(a) HAS using HTTP/1.1.

(b) The k-push method using HTTP/2 [57].

Fig. 14. HAS using HTTP/1.1 versus the k-push method using HTTP/2.

be split into a subset of bitstreams instead of different bitrate levels, and thus, the video segments can be encoded at different SVC qualities (temporal, spatial, SNR). Using this mechanism, a HAS client can incrementally improve the quality of a segment by fetching additional bitstreams or layers depending on the dynamics of the available bandwidth. One key difference when using SVC with DASH is that a client may have to download multiple segments (*i.e.,* base and enhancement layers) for one playback epoch, unlike in the case of non-scalable video. Dayananda and Swaminathan [164] investigated the gain of SHVC in HAS, and they found that it could result in bitrate savings but at the price of increased encoding overhead due to scalability. Interestingly, MPEG's exploration towards a future video coding format with capabilities beyond HEVC initially suggested having scalability as a built-in feature, but that has been withdrawn from the final call for proposals [165], and thus, is not considered in Versatile Video Coding (VVC) [23].

B. HTTP/2 and QUIC-Based Streaming

Google initially developed SPDY [166], which eventually led to the specification of HTTP/2 [167], and also developed QUIC [168], which, with HTTP/2, addresses the latency and head-of-line (HOL) blocking issues that were inherent to HTTP/1.1 over TCP. Both HTTP/2 and QUIC may have an impact on the HAS performance [79], [169]–[171].

1) HTTP/2: HTTP/2 is used over a single *persistent* TCP connection (with pipelining support) between the client and server comprising multiple streams in a full duplex model with advanced features such as frame exchange, request prioritization, header field compression, and server push. A first evaluation has been conducted by Mueller *et al.* [169], which shows that HTTP/2 can achieve a similar performance compared to HTTP/1.1 (with pipelined persistent connections enabled). Wei and Swaminathan [57] used the HTTP/2 server push feature and introduced k-push to reduce both live latency and the number of segment requests. In k-push, the client sends one request to the server every k segments indicating the number of segments (k) to be pushed to the client. The server responds by pushing each segment consecutively as soon as it becomes available, but all at the same requested bitrate level (see Fig. 14). Xiao *et al.* [172] further evaluated the k-push scheme and showed that it deteriorated network adaptability, since its gains diminish as k increases and it led to the "over-push" problem where network resources are wasted due to video abandonment by the viewers. Thus, the authors proposed adaptive-push to overcome k-push's limitations, which uses the same principle as its predecessor but selects k adaptively. In both k-push and adaptive-push, the client can implement various rate adaptation algorithms. Cherif *et al.* [173] also used HTTP/2 server push to implement a fast startup where segments were initially pushed to the client upon receiving a request for the MPD. As the client would typically be unaware of the initial bandwidth conditions, the authors suggested using a WebSocket connection over HTTP/2 to exchange various status messages including bandwidth estimation information. Finally, an overview of HTTP/2-based methods to improve the live experience of HAS has been presented in [174], which includes *(1)* stream termination, *(2)* request/response multiplexing and stream prioritization, and *(3)* server push. It provides a detailed analysis of HTTP/2-based QoE-improvement methods including a comprehensive evaluation.

2) QUIC: QUIC is a UDP-based secure transport layer protocol that aims to speed up the connections, reduce latencies, enable congestion and flow control, allow multiple (multiplexed/pipelined) data connections (*e.g.,* HTTP request/response) over the same UDP connection without HOL blocking, and UDP connection migration with Forward Error Correction (FEC). QUIC has been evaluated in the context of HAS by Timmerer and Bertoni [170] and the results show a similar adaptation performance of HTTP/2 over TCP, HTTP/2 over SSL, HTTP/1.1 over QUIC and SPDY over QUIC. The experimental results reported that QUIC introduces around 10% more overhead than TCP at low bitrates. Also, the bandwidth utilization decreases when the round-trip time (RTT) increases, but it remains high and stable around 87%. A similar evaluation was conducted by Bhat *et al.* [171], which revealed that bitrate adaptation schemes deployed on top of QUIC do not show a performance increase unless the existing schemes are properly adjusted to be used in conjunction with QUIC. Other evaluations of QUIC have focused on generic traffic patterns such as regular Web sites [175], [176] without providing details on HAS.

C. Immersive Media Streaming

Immersive media streaming and specifically virtual reality (VR)/360-degree video streaming is nowadays gaining significant attention from both academia and industry due to the increasing availability of 360-degree cameras and head mounted displays (HMD). VR applications range from 3D video gaming to 360-degree video streaming and teleimmersion. In this survey, we highlight the use of HAS in 360-degree video streaming.

1) Characteristics of 360-Degree Videos: 360-degree videos are recorded using multiple specialized high-resolution cameras that capture a sphere around the user. The resulting video is typically stitched and mapped onto a 2D plane using various projection formats due to a lack of coding tools for the spherical domain. At the client side, the 2D plane is mapped back on a surface mesh and rendered based on the device capabilities. Characteristically, they allow users to freely navigate within the media presentation but only a fraction of the actual content is presented to the user at any given point in time. This is referred to as the *viewport*, or field of view. Considering the high resolution nature of the full spherical content, the amount of data to be streamed may be significantly higher than the one for conventional, non-360-degree videos.

2) Adaptive Streaming Challenges: Most adaptive streaming schemes for 360-degree videos merely adopt traditional non-360-degree video delivery schemes. The entire 360-degree scene is adaptively delivered without taking the user's viewport into account. For example, the content outside the user's current viewport is delivered at the same quality as the content within the user's viewport, wasting bandwidth, and thus, network resources. Viewport-adaptive [177] and tile-based adaptive streaming techniques [178], [179] are currently suggested in the literature to overcome this disadvantage. The former provides pre-encoded versions of a given viewport based on the user's device orientation, which requires additional content versions to be prepared, stored, and distributed within the delivery network. The latter uses the tiling feature available in modern video codecs (*e.g.,* in HEVC, VP9, and AV1) that enables spatial segmentation of videos. Each tile can be projected in different representations to allow for quality adaptation. However, requesting each tile individually may increase the number of requests tremendously, which could be addressed by the server push feature of HTTP/2 as suggested in [180]. A number of open issues are also discussed by Graf *et al.* [179] ranging from encoding/streaming issues to QoE.

3) Standardization: Several standardization bodies and industry forums have started working towards achieving interoperability between different VR systems. An overview is provided in [181]. MPEG's efforts to standardize the storage and delivery formats for 360-degree video content is specified in the Omnidirectional Media Format (OMAF) standard [182]. OMAF describes the content processing architecture, projection and packaging formats, streaming approaches and DASH integration of 360-degree videos [183].

D. HAS Datasets

In the past, a great number of DASH datasets has emerged. The first DASH dataset was released by Lederer *et al.* [19] and comprises various genres (*i.e.,* animation, sport, movie), encoded using up to 20 representations (up to 1080p resolution), and different segment lengths (*i.e.,* 1, 2, 4, 6, 10, and 15 seconds). Additionally, for some representations per frame PSNR values are provided. Initial evaluations of the dataset provide recommendations for an optimal segment length based on the coding efficiency (*i.e.,* 4s) and the influence of enabled versus disabled persistent connections.

A distributed DASH dataset has been released by Lederer *et al.* [184], which distributes the dataset across multiple locations and utilizes multiple *BaseURL* elements within the media presentation description (MPD). It can be used to simulate different content distribution network (CDN) locations and bitstream switching across multiple CDNs.

Le Feuvre *et al.* [185] provide an ultra high definition (UHD) HEVC DASH dataset targeting UHD services (*i.e.,* resolutions up to 3840x2160, framerate up to 60 fps, and up to 10 bpp) using HEVC, which is the major difference compared to previously proposed datasets. Kreuzberger *et al.* [30] provides a DASH dataset focusing on scalable video coding (SVC) and experimenting with in-network adaptation in named data networks and information-centric networking, respectively. Unfortunately, support for SVC in end user devices is still limited. Quinlan *et al.* [186] propose a dataset comprising AVC and HEVC for the evaluation of DASH systems.

Finally, Zabrovskiy *et al.* [187] provide a multi-codec DASH dataset comprising multiple state-of-the-art as well as emerging video codecs, *i.e.,* AVC, HEVC, VP9, and AV1 to enable interoperability testing and allow for experimenting with adaptation strategies of DASH clients supporting multiple video codecs. A similar dataset is provided by Quinlan and Sreenan [188] focusing on AVC and HEVC for UHD (4K) resolutions.

VI. Conclusion

Since the emergence of HTTP adaptive streaming (HAS), many bitrate adaption schemes have been proposed. Each is trying to address certain HAS-related problems and striving to achieve a set of goals. In fact, most state-of-the-art schemes share a common main objective, which is to improve viewer

QoE. In this survey, we examined a set of well-known schemes and heuristics for their applicability.

Firstly, we classified the bitrate adaptation schemes into four main categories, namely, client-based, server-based, network-assisted and hybrid. In a client-based scheme, the client strives to optimize the viewer QoE individually and considers one of the many heuristics based on the available bandwidth, playback buffer size, segment size, and duration. Server-based schemes, in contrast, do not require any cooperation from the clients, and they use a server traffic shaping mechanism. In network-assisted schemes, the clients use information coming from in-network devices, like proxies, together with their own observations for bitrate adaptation. Finally, the hybrid solutions consist of many entities like clients, central managers, servers, and network devices that are involved in the bitrate decision process.

Secondly, we offered a description of each scheme by presenting the problems they are trying to solve, their goals, findings, main components and critical acclaims. Although the described schemes in each category provide noteworthy benefits and efficiency in some specific network characteristics, many shared challenges exist in every category, especially when multiple clients compete for the shared bandwidth:

- Client-based schemes likely suffer from HAS stability issues and QoE variations due to the HAS' ON-OFF pattern. These issues are aggravated when the number of geographically-distributed clients keeps growing.
- Server-based schemes introduce overhead and complexity, limiting the system scalability with the increasing number of clients.
- Network-assisted and hybrid adaptation schemes use centralized entities to assist the clients in their decisions, improve the viewer QoE, and avoid HAS scalability issues. However, they are difficult to deploy over the fully decentralized nature of real-world network infrastructures and they do not support large-scale deployments where many HAS players are geographically distributed.

Thirdly, we provided a comparison between the surveyed schemes in terms of a set of QoE and networking aspects. Our comparison may help researchers in the area of adaptive streaming where it offers a general consistent framework that can formally evaluate and compare different bitrate adaptation logic categories, and test the efficiency of their components. Finally, we concluded the survey by a general discussion on the recent developments in HAS systems, such as the use of HTTP/2 and QUIC as well as HAS of VR content.

In general, certain limitations still exist when conducting a comprehensive survey. The lack of standardized benchmarks and frameworks (*i.e.,* datasets, test conditions and QoE metrics) makes any performance comparison a difficult task. For example, a fair comparison between client-based adaptation schemes in terms of performance (*i.e.,* resource utilization) and QoE (*i.e.,* video stalls, stabilization, quality oscillations), requires that they undergo similar experimentation configuration, including a unified bandwidth trace, certain networking setups and similar device capabilities. The surveyed schemes may have perform well under certain conditions, but they all use various heuristics that broadly relate to specific settings, operating regimes (*i.e.,* different network environments, chunk sizes, content types, *etc.*), and may require parameter tuning. A common set of test conditions might reveal significantly different results than the ones reported in the original papers. In the broad area of adaptive streaming, there are many open challenges and issues that need more attention:

- Understanding the main factors that degrade the viewer QoE through subjective and objective tests; then, designing a standardized QoE function.
- Designing placement algorithms for CDN, proxies and SDN controllers.
- Understanding the trade-off between content-aware encoding versus content-aware streaming (generating variable bitrate encoded segments is easy, but streaming them is not).
- Designing a robust solution that achieves fair resource sharing among concurrent HAS clients when they compete in a bottleneck network.
- Understanding multi-path benefits and adding its capabilities to HAS delivery systems.
- Studying the interaction between HAS and non-HAS traffic, and its impact on the QoE.
- Mixing client-based and hybrid solutions without introducing extra overhead.
- Providing a solution to deliver 360-degree videos that reduces bandwidth consumption while not hampering the QoE.
- Leveraging machine learning and deep learning techniques to analyze and classify encrypted HAS traffic, which can help monitor and mitigate QoE impairments.

Acknowledgment

The authors would like to thank Prof. Saad Harous for his valuable feedback.

References

[1] "Cisco visual networking index: Forecast and methodology, 2016–2021," San Jose, CA, USA, Cisco Syst., Inc., White Paper, 2017.
[2] *Real-Time Messaging Protocol(RTMP)*, Adobe, San Jose, CA, USA, 2014. Accessed: Nov. 21, 2017. [Online]. Available: http://www.adobe.com/devnet/rtmp.html
[3] V. Jacobson, R. Frederick, S. Casner, and H. Schulzrinne. (2014). *Real-Time Transport Protocol (RTP)*. Accessed: Nov. 21, 2017. [Online]. Available: https://www.ietf.org/rfc/rfc3550.txt
[4] H. Schulzrinne. (2016). *Real Time Streaming Protocol Version 2.0*. Accessed: Nov. 21, 2017. [Online]. Available: https://tools.ietf.org/html/rfc7826
[5] J. Goldberg, M. Westerlund, and T. Zeng. (2014). *A Network Address Translator (NAT) Traversal Mechanism for Media Controlled by the Real-Time Streaming Protocol (RTSP)*. Accessed: Nov. 21, 2017. [Online]. Available: https://tools.ietf.org/html/rfc7825
[6] L. Popa, A. Ghodsi, and I. Stoica, "HTTP as the narrow waist of the future Internet," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw. (Hotnets-IX)*, 2010, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/1868447.1868453
[7] X. Liu *et al.*, "A case for a coordinated Internet video control plane," in *Proc. ACM SIGCOMM Conf. Appl. Technol. Archit. Protocols Comput. Commun. (SIGCOMM)*, 2012, pp. 359–370. [Online]. Available: http://doi.acm.org/10.1145/2342356.2342431
[8] *Microsoft Smooth Streaming*, Microsoft, Redmond, WA, USA, 2015. Accessed: Nov. 21, 2017. [Online]. Available: http://www.iis.net/downloads/microsoft/smooth-streaming

[9] *Apple HTTP Live Streaming*, Apple, Cupertino, CA, USA, 2015. Accessed: Nov. 21, 2017. [Online]. Available: https://developer.apple.com/streaming/

[10] *Adobe HTTP Dynamic Streaming*, Adobe, San Jose, CA, USA, 2015. Accessed: Nov. 21, 2017. [Online]. Available: http://www.adobe.com/products/hds-dynamic-streaming.html

[11] Akamai. (2015). *Akamai HD*. Accessed: Nov. 21, 2017. [Online]. Available: https://www.akamai.com/us/en/resources/live-video-streaming.jsp

[12] C. Timmerer. (2012). *HTTP Streaming of MPEG Media*. Accessed: Nov. 21, 2017. [Online]. Available: https://multimediacommunication.blogspot.co.at/2010/05/http-streaming-of-mpeg-media.html

[13] Dash Industry Forum. (2017). *DASH-264 JavaScript Reference Client*. Accessed: Nov. 21, 2017. [Online]. Available: http://dashif.org/reference/players/javascript/index.html

[14] B. Rainer, S. Lederer, C. Müller, and C. Timmerer, "A seamless Web integration of adaptive HTTP streaming," in *Proc. 20th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2012, pp. 1519–1523.

[15] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1842–1866, 3rd Quart., 2017.

[16] E. Thomas, M. O. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey, "Enhancing MPEG DASH performance via server and network assistance," *SMPTE Motion Imag. J.*, vol. 126, no. 1, pp. 22–27, 2017.

[17] X. Wang, "Network-assistance and server management in adaptive streaming on the Internet," in *Proc. W3C Web TV Workshop*, Munich, Germany, 2014. [Online]. Available: https://www.w3.org/2013/10/tv-workshop/papers/webtv4_submission_17.pdf

[18] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[19] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over HTTP dataset," in *Proc. 3rd Multimedia Syst. Conf. (MMSys)*, 2012, pp. 89–94. [Online]. Available: http://doi.acm.org/10.1145/2155555.2155570

[20] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[21] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[22] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of SHVC: Scalable extensions of the high efficiency video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 20–34, Jan. 2016.

[23] C. Timmerer, "MPEG column: 122nd MPEG meeting in San Diego, CA, USA," *SIGMultimedia Rec.*, vol. 10, no. 2, p. 6, Jun. 2018.

[24] J. De Cock, A. Mavlankar, A. Moorthy, and A. Aaron, "A large-scale video codec comparison of x264, x265 and libvpx for practical VOD applications," in *Proc. SPIE*, vol. 9971. San Diego, CA, USA, 2016, Art. no. 997116.

[25] M. Seufert *et al.*, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, 1st Quart., 2015.

[26] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptive video players over HTTP," *Signal Process. Image Commun.*, vol. 27, no. 4, pp. 271–287, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596511001159

[27] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when HTTP adaptive streaming players compete for bandwidth?" in *Proc. 22nd Int. Workshop Netw. Oper. Syst. Support Digit. Audio Video (NOSSDAV)*, 2012, pp. 9–14. [Online]. Available: http://doi.acm.org/10.1145/2229087.2229092

[28] S. Bae, D. Jang, and K. Park, "Why Is HTTP adaptive streaming so hard?" in *Proc. 6th Asia–Pac. Workshop Syst. (APSys)*, 2015, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/2797022.2797031

[29] G. Cermak, M. Pinson, and S. Wolf, "The relationship among video quality, screen resolution, and bit rate," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 258–262, Jun. 2011.

[30] C. Kreuzberger, D. Posch, and H. Hellwagner, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP," in *Proc. 6th ACM Multimedia Syst. Conf. (MMSys)*, 2015, pp. 213–218. [Online]. Available: http://doi.acm.org/10.1145/2713168.2713193

[31] Z. Duanmu, K. Zeng, K. Ma, A. Rehman, and Z. Wang, "A quality-of-experience index for streaming video," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 1, pp. 154–166, Feb. 2017.

[32] Z. Li *et al.*, "Streaming video over HTTP with consistent quality," in *Proc. 5th ACM Multimedia Syst. Conf. (MMSys)*, 2014, pp. 248–258. [Online]. Available: http://doi.acm.org/10.1145/2557642.2557658

[33] L. Yu, T. Tillo, and J. Xiao, "QoE-driven dynamic adaptive video streaming strategy with future information," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 523–534, Sep. 2017.

[34] T. Hossfeld *et al.*, "Initial delay vs. interruptions: Between the devil and the deep blue sea," in *Proc. 4th Int. Workshop Qual. Multimedia Exp.*, Jul. 2012, pp. 1–6.

[35] T. De Pessemier, K. De Moor, W. Joseph, L. De Marez, and L. Martens, "Quantifying the influence of rebuffering interruptions on the user's quality of experience during mobile video watching," *IEEE Trans. Broadcast.*, vol. 59, no. 1, pp. 47–61, Mar. 2013.

[36] Y. Qi and M. Dai, "The effect of frame freezing and frame skipping on video quality," in *Proc. Int. Conf. Intell. Inf. Hiding Multimedia*, Dec. 2006, pp. 423–426.

[37] D. C. Robinson, Y. Jutras, and V. Craciun, "Subjective video quality assessment of HTTP adaptive streaming technologies," *Bell Lab. Tech. J.*, vol. 16, no. 4, pp. 5–23, Mar. 2012, doi: 10.1002/bltj.20531.

[38] R. Hamberg and H. de Ridder, "Time-varying image quality: Modeling the relation between instantaneous and overall quality," *SMPTE J.*, vol. 108, no. 11, pp. 802–811, Nov. 1999.

[39] N. Cranley, P. Perry, and L. Murphy, "User perception of adapting video quality," *Int. J. Human–Comput. Stud.*, vol. 64, no. 8, pp. 637–647, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581905002028

[40] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.

[41] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, Jun. 2008.

[42] A. Rehman, K. Zeng, and Z. Wang, "Display device-adapted video quality-of-experience assessment," in *Proc. SPIE Human Vis. Electron. Imag.*, vol. 9394. San Francisco, CA, USA, 2015, Art. no. 939406.

[43] K. ur Rehman Laghari, O. Issa, F. Speranza, and T. H. Falk, "Quality-of-experience perception for video streaming services: Preliminary subjective and objective results," in *Proc. Asia–Pac. Signal Inf. Process. Assoc. Annu. Summit Conf.*, Dec. 2012, pp. 1–9.

[44] B. Rainer, S. Petscharnig, C. Timmerer, and H. Hellwagner, "Statistically indifferent quality variation: An approach for reducing multimedia distribution cost for adaptive video streaming services," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 849–860, Apr. 2017.

[45] A. Balachandran *et al.*, "Developing a predictive model of quality of experience for Internet video," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 339–350, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2534169.2486025

[46] M. Montagud, F. Boronat, H. Stokking, and R. van Brandenburg, "Inter-destination multimedia synchronization: Schemes, use cases and standardization," *Multimedia Syst.*, vol. 18, no. 6, pp. 459–482, 2012.

[47] B. Rainer and C. Timmerer, "Self-organized inter-destination multimedia synchronization for adaptive media streaming," in *Proc. 22nd ACM Int. Conf. Multimedia (MM)*, 2014, pp. 327–336. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654938

[48] B. Rainer, S. Petscharnig, and C. Timmerer, "Merge and forward: Self-organized inter-destination multimedia synchronization," in *Proc. 6th ACM Multimedia Syst. Conf. (MMSys)*, 2015, pp. 77–80. [Online]. Available: http://doi.acm.org/10.1145/2713168.2713185

[49] B. Rainer, S. Petscharnig, C. Timmerer, and H. Hellwagner, "Is one second enough? Evaluating QoE for inter-destination multimedia synchronization using human computation and crowdsourcing," in *Proc. 7th Int. Workshop Qual. Multimedia Exp. (QoMEX)*, 2015, pp. 1–6.

[50] T. Stockhammer, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst. (MMSys)*, 2011, pp. 133–144. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943572

[51] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst. (MMSys)*, 2011, pp. 169–174. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943575

[52] C. Liu, I. Bouazizi, M. M. Hannuksela, and M. Gabbouj, "Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network," *Signal Process. Image Commun.*, vol. 27, no. 4, pp. 288–311, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596511001135

[53] Z. Li *et al.*, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 719–733, Apr. 2014.

[54] X. Xie, X. Zhang, S. Kumar, and L. E. Li, "piStream: Physical layer informed adaptive video streaming over LTE," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2015, pp. 413–425. [Online]. Available: http://doi.acm.org/10.1145/2789168.2790118

[55] T. Andelin, V. Chetty, D. Harbaugh, S. Warnick, and D. Zappala, "Quality selection for dynamic adaptive streaming over HTTP with scalable video coding," in *Proc. 3rd Multimedia Syst. Conf. (MMSys)*, 2012, pp. 149–154. [Online]. Available: http://doi.acm.org/10.1145/2155555.2155580

[56] M. Xiao, V. Swaminathan, S. Wei, and S. Chen, "DASH2M: Exploring HTTP/2 for Internet streaming to mobile devices," in *Proc. ACM Multimedia Conf. (MM)*, 2016, pp. 22–31. [Online]. Available: http://doi.acm.org/10.1145/2964284.2964313

[57] S. Wei and V. Swaminathan, "Low latency live video streaming over HTTP 2.0," in *Proc. Netw. Oper. Syst. Support Digit. Audio Video Workshop*, 2014, p. 37. [Online]. Available: http://doi.acm.org/10.1145/2578260.2578277

[58] K. Miller, A.-K. Al-Tamimi, and A. Wolisz, "QoE-based low-delay live streaming using throughput predictions," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 1, pp. 1–4, Oct. 2016. [Online]. Available: http://doi.acm.org/10.1145/2990505

[59] J. Hao, R. Zimmermann, and H. Ma, "GTube: Geo-predictive video streaming over HTTP in mobile environments," in *Proc. 5th ACM Multimedia Syst. Conf. (MMSys)*, 2014, pp. 259–270. [Online]. Available: http://doi.acm.org/10.1145/2557642.2557647

[60] H. Riiser, H. S. Bergsaker, P. Vigmostad, P. Halvorsen, and C. Griwodz, "A comparison of quality scheduling in commercial adaptive HTTP streaming solutions on a 3G network," in *Proc. 4th Workshop Mobile Video (MoVid)*, 2012, pp. 25–30. [Online]. Available: http://doi.acm.org/10.1145/2151677.2151684

[61] J. Yao, S. S. Kanhere, and M. Hassan, "Improving QoS in high-speed mobility using bandwidth maps," *IEEE Trans. Mobile Comput.*, vol. 11, no. 4, pp. 603–617, Apr. 2012.

[62] J. Yao, S. S. Kanhere, I. Hossain, and M. Hassan, "Empirical evaluation of HTTP adaptive streaming under vehicular mobility," in *NETWORKING 2011*, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds. Heidelberg, Germany: Springer, 2011, pp. 92–105.

[63] V. Singh, J. Ott, and I. D. D. Curcio, "Predictive buffering for streaming video in 3G networks," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Jun. 2012, pp. 1–10.

[64] B. Taani and R. Zimmermann, "Spatio-temporal analysis of bandwidth maps for geo-predictive video streaming in mobile environments," in *Proc. ACM Multimedia Conf. (MM)*, 2016, pp. 888–897. [Online]. Available: http://doi.acm.org/10.1145/2964284.2964333

[65] C. Mueller, S. Lederer, R. Grandl, and C. Timmerer, "Oscillation compensating dynamic adaptive streaming over HTTP," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jun. 2015, pp. 1–6.

[66] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, Aug. 2014. [Online]. Available: http://doi.acm.org/10.1145/2740070.2626296

[67] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in *Proc. IEEE INFOCOM 35th Annu. Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[68] C. Sieber, T. Hoßfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and user-centric comparison of a novel adaptation logic for DASH with SVC," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, May 2013, pp. 1318–1323.

[69] P. K. Yadav, A. Shafiei, and W. T. Ooi, "QUETRA: A queuing theory approach to DASH rate adaptation," in *Proc. ACM Multimedia Conf. (MM)*, 2017, pp. 1130–1138. [Online]. Available: http://doi.acm.org/10.1145/3123266.3123390

[70] R. Huysegems, B. De Vleeschauwer, T. Wu, and W. Van Leekwijck, "SVC-based HTTP adaptive streaming," *Bell Labs Tech. J.*, vol. 16, no. 4, pp. 25–41, Mar. 2012.

[71] C. Mueller. (2015). *Microsoft Smooth Streaming*. Accessed: Nov. 21, 2017. [Online]. Available: https://bitmovin.com/microsoft-smooth-streaming-mss/

[72] R. Pantos and W. May. (2017). *HTTP Live Streaming*. Accessed: Dec. 20, 2017. [Online]. Available: https://www.ietf.org/rfc/rfc8216.txt

[73] Apple. (2016). *QuickTime*. Accessed: Nov. 21, 2017. [Online]. Available: http://www.apple.com/sg/quicktime/

[74] (2015). *Simplified Adaptive Video Streaming: Announcing Support for HLS and DASH in Windows 10*. Accessed: Dec. 20, 2017. [Online]. Available: https://goo.gl/gZM3mQ

[75] *Supported Media Formats | Android Developers*. Accessed: Dec. 20, 2017. [Online]. Available: https://developer.android.com/guide/topics/media/media-formats.html

[76] *ActionScript*, Adobe, San Jose, CA, USA, 2016. Accessed: Nov. 21, 2017. [Online]. Available: http://www.adobe.com/devnet/actionscript.html

[77] T. Cloonan and J. Allen, "Competitive analysis of adaptive video streaming implementations," in *Proc. SCTE Cable-Tec Expo Tech. Workshop*, 2011, pp. 1–34.

[78] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the Internet: Approaches and directions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 282–300, Mar. 2001.

[79] C. Müller, S. Lederer, and C. Timmerer, "An evaluation of dynamic adaptive streaming over HTTP in vehicular environments," in *Proc. ACM 4th Workshop Mobile Video (MoVid)*, 2012, pp. 37–42. [Online]. Available: http://doi.acm.org/10.1145/2151677.2151686

[80] Bitmovin. (2015). *MPEG-DASH vs. Commercial Players*. Accessed: Nov. 21, 2017. [Online]. Available: http://www.goo.gl/TmazZ8

[81] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 325–338, Aug. 2015. [Online]. Available: http://doi.acm.org/10.1145/2829988.2787486

[82] C. Zhou, X. Zhang, L. Huo, and Z. Guo, "A control-theoretic approach to rate adaptation for dynamic HTTP streaming," in *Proc. Vis. Commun. Image Process.*, Nov. 2012, pp. 1–6.

[83] A. Sobhani, A. Yassine, and S. Shirmohammadi, "A video bitrate adaptation and prediction mechanism for HTTP adaptive streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 2, p. 18, Mar. 2017. [Online]. Available: http://doi.acm.org/10.1145/3052822

[84] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: A client-side controller for dynamic adaptive streaming over HTTP (DASH)," in *Proc. 20th Int. Packet Video Workshop*, Dec. 2013, pp. 1–8.

[85] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. New York, NY, USA: Macmillan, 2013.

[86] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *Proc. 19th Int. Packet Video Workshop (PV)*, May 2012, pp. 173–178.

[87] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," in *Proc. ACM 8th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2012, pp. 97–108. [Online]. Available: http://doi.acm.org/10.1145/2413176.2413189

[88] C. Zhou, C. Lin, X. Zhang, and Z. Guo, "TFDASH: A fairness, stability, and efficiency aware rate control approach for multiple clients over DASH," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2017.2771246.

[89] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proc. ACM 8th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2012, pp. 109–120. [Online]. Available: http://doi.acm.org/10.1145/2413176.2413190

[90] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004, doi: 10.1023/B:STCO.0000035301.49549.88.

[91] C. Wang, A. Rizk, and M. Zink, "SQUAD: A spectrum-based quality adaptation for dynamic adaptive streaming over HTTP," in *Proc. ACM 7th Int. Conf. Multimedia Syst. (MMSys)*, 2016, p. 1. [Online]. Available: http://doi.acm.org/10.1145/2910017.2910593

[92] D. Havey, R. Chertov, and K. Almeroth, "Receiver driven rate adaptation for wireless multimedia applications," in *Proc. ACM 3rd Multimedia Syst. Conf. (MMSys)*, 2012, pp. 155–166. [Online]. Available: http://doi.acm.org/10.1145/2155555.2155582

[93] P. Juluri, V. Tamarapalli, and D. Medhi, "SARA: Segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 1765–1770.

[94] A. Beben, P. Wiśniewski, J. M. Batalla, and P. Krawiec, "ABMA+: Lightweight and efficient algorithm for HTTP adaptive streaming," in *Proc. ACM 7th Int. Conf. Multimedia Syst. (MMSys)*, 2016, p. 2. [Online]. Available: http://doi.acm.org/10.1145/2910017.2910596

[95] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "Want to play DASH? A game theoretic approach for adaptive streaming over HTTP," in *Proc. ACM MMSys*, 2018, pp. 13–26.

[96] R. B. Myerson, *Game Theory*. New York, NY, USA: Harvard Univ. Press, 2013.

[97] M. Xing, S. Xiang, and L. Cai, "A real-time adaptive algorithm for video streaming over multiple wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 795–805, Apr. 2014.

[98] A. Bokani, M. Hassan, S. Kanhere, and X. Zhu, "Optimizing HTTP-based adaptive streaming in vehicular environment using Markov decision process," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2297–2309, Dec. 2015.

[99] S. Petrangeli, M. Claeys, S. Latré, J. Famaey, and F. De Turck, "A multi-agent *Q*-learning-based framework for achieving fairness in HTTP adaptive streaming," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, May 2014, pp. 1–9.

[100] F. Chiariotti, S. D'Aronco, L. Toni, and P. Frossard, "Online learning adaptation strategy for DASH clients," in *Proc. ACM 7th Int. Conf. Multimedia Syst. (MMSys)*, 2016, p. 8. [Online]. Available: http://doi.acm.org/10.1145/2910017.2910603

[101] C. Zhou, C.-W. Lin, and Z. Guo, "mDASH: A Markov decision-based rate adaptation approach for dynamic HTTP streaming," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 738–751, Apr. 2016.

[102] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with Pensieve," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, 2017, pp. 197–210. [Online]. Available: http://doi.acm.org/10.1145/3098822.3098843

[103] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-DASH: A deep *Q*-learning framework for DASH video streaming," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 703–718, Dec. 2017.

[104] Y. Li, "Deep reinforcement learning: An overview," *CoRR*, vol. abs/1701.07274, 2017. [Online]. Available: http://arxiv.org/abs/1701.07274

[105] J. K. Rowling, *Harry Potter and the Goblet of Fire*. London, U.K.: Bloomsbury, 2000.

[106] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen, "Server-based traffic shaping for stabilizing oscillating adaptive streaming players," in *Proc. 23rd ACM Workshop Netw. Oper. Syst. Support Digit. Audio Video (NOSSDAV)*, 2013, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/2460782.2460786

[107] R. Houdaille and S. Gouache, "Shaping HTTP adaptive streams for a better user experience," in *Proc. ACM 3rd Multimedia Syst. Conf. (MMSys)*, 2012, pp. 1–9. [Online]. Available: http://doi.acm.org/10.1145/2155555.2155557

[108] A. Detti, B. Ricci, and N. Blefari-Melazzi, "Tracker-assisted rate adaptation for MPEG DASH live streaming," in *Proc. IEEE INFOCOM 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[109] L. De Cicco, S. Mascolo, and V. Palmisano, "Feedback control for adaptive live video streaming," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst. (MMSys)*, 2011, pp. 145–156. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943573

[110] J. Bruneau-Queyreix, M. Lacaud, D. Negru, J. M. Batalla, and E. Borcoci, "MS-stream: A multiple-source adaptive streaming solution enhancing consumer's perceived quality," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 427–434.

[111] E. Thomas *et al.*, "Applications and deployments of server and network assisted DASH (SAND)," in *Proc. Int. Broadcast. Conv. Conf. (IBC)*, Amsterdam, The Netherlands, 2016, p. 22.

[112] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: A QoE-aware DASH system," in *Proc. ACM 3rd Multimedia Syst. Conf. (MMSys)*, 2012, pp. 11–22. [Online]. Available: http://doi.acm.org/10.1145/2155555.2155558

[113] N. Bouten *et al.*, "QoE-driven in-network optimization for adaptive video streaming based on packet sampling measurements," *Comput. Netw.*, vol. 81, pp. 96–115, Apr. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128615000468

[114] V. Krishnamoorthi, N. Carlsson, E. Halepovic, and E. Petajan, "BUFFEST: Predicting buffer conditions and real-time requirements of HTTP(S) adaptive streaming clients," in *Proc. 8th ACM Multimedia Syst. Conf. (MMSys)*, 2017, pp. 76–87. [Online]. Available: http://doi.acm.org/10.1145/3083187.3083193

[115] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[116] S. D'Aronco, L. Toni, and P. Frossard, "Price-based controller for utility-aware HTTP adaptive streaming," *IEEE MultiMedia*, vol. 24, no. 2, pp. 20–29, Apr./Jun. 2017.

[117] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "QoE-driven rate adaptation heuristic for fair adaptive video streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 2, p. 28, 2016.

[118] V. Joseph and G. de Veciana, "NOVA: QoE-driven optimization of DASH-based video delivery in networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2014, pp. 82–90.

[119] R. M. Gray and R. Gray, *Probability, Random Processes, and Ergodic Properties*. New York, NY, USA: Springer, 1988.

[120] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks," in *Proc. ACM 19th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2013, pp. 389–400. [Online]. Available: http://doi.acm.org/10.1145/2500423.2500433

[121] A. El Essaili *et al.*, "Quality-of-experience driven adaptive HTTP media delivery," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 2480–2485.

[122] J. W. Kleinrouweler, S. Cabrero, R. van der Mei, and P. Cesar, "Modeling stability and bitrate of network-assisted HTTP adaptive streaming players," in *Proc. 27th Int. Teletraffic Congr. (ITC)*, 2015, pp. 177–184, doi: 10.1109/ITC.2015.28

[123] V. Ramamurthi and O. Oyman, "Video-QoE aware radio resource allocation for HTTP adaptive streaming," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1076–1081.

[124] V. Ramamurthi, O. Oyman, and J. Foerster, "Video-QoE aware resource management at network core," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2014, pp. 1418–1423.

[125] B. Han, F. Qian, L. Ji, V. Gopalakrishnan, and N. Bedminster, "MP-DASH: Adaptive video streaming over preference-aware multipath," in *Proc. ACM 12th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2016, pp. 129–143. [Online]. Available: http://doi.acm.org/10.1145/2999572.2999606

[126] Z. Yan, J. Xue, and C. W. Chen, "Prius: Hybrid edge cloud and client adaptation for HTTP adaptive streaming in cellular networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 1, pp. 209–222, Jan. 2017.

[127] A. H. Zahran, J. J. Quinlan, K. K. Ramakrishnan, and C. J. Sreenan, "SAP: Stall-aware pacing for improved DASH video experience in cellular networks," in *Proc. 8th ACM Multimedia Syst. Conf. (MMSys)*, 2017, pp. 13–26. [Online]. Available: http://doi.acm.org/10.1145/3083187.3083199

[128] T. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.

[129] M. D. F. De Grazia *et al.*, "QoE multi-stage machine learning for dynamic video streaming," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 1, pp. 146–161, Mar. 2018.

[130] G. Urvoy-Keller, "On the stationarity of TCP bulk data transfers," in *Proc. 6th Int. Conf. Passive Active Netw. Meas. (PAM)*, 2005, pp. 27–40, doi: 10.1007/978-3-540-31966-5_3.

[131] Z. Akhtar *et al.*, "Oboe: Auto-tuning video ABR algorithms to network conditions," in *Proc. SIGCOMM Comput. Commun. Rev.*, Aug. 2018. [Online]. Available: https://engineering.purdue.edu/∼isl/papers/sigcomm18-final128.pdf

[132] P. Georgopoulos, M. Broadbent, A. Farshad, B. Plattner, and N. Race, "Using software defined networking to enhance the delivery of video-on-demand," *Comput. Commun.*, vol. 69, pp. 79–87, Sep. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366415002315

[133] G. Cofano *et al.*, "Design and experimental evaluation of network-assisted strategies for HTTP adaptive streaming," in *Proc. ACM 7th Int. Conf. Multimedia Syst. (MMSys)*, 2016, p. 3. [Online]. Available: http://doi.acm.org/10.1145/2910017.2910597

[134] D. Bhat, A. Rizk, M. Zink, and R. Steinmetz, "Network assisted content distribution for adaptive bitrate video streaming," in *Proc. 8th ACM Multimedia Syst. Conf. (MMSys)*, 2017, pp. 62–75. [Online]. Available: http://doi.acm.org/10.1145/3083187.3083196

[135] A. Seema, L. Schwoebel, T. Shah, J. Morgan, and M. Reisslein, "WVSNP-DASH: Name-based segmented video streaming," *IEEE Trans. Broadcast.*, vol. 61, no. 3, pp. 346–355, Sep. 2015.

[136] D. E. Knuth, "Backus normal form vs. Backus Naur form," *Commun. ACM*, vol. 7, no. 12, pp. 735–736, Dec. 1964. [Online]. Available: http://doi.acm.org/10.1145/355588.365140

[137] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagner, "Adaptive multimedia streaming in information-centric networks," *IEEE Netw.*, vol. 28, no. 6, pp. 91–96, Nov. 2014.

[138] C. Westphal *et al.*, "Adaptive video streaming over information-centric networking (ICN), Internet Eng. Task Force, Fremont, CA, USA, Rep. RFC 7933, Aug. 2016. [Online]. Available: http://www.ietf.org/rfc/rfc7933.txt

[139] B. Rainer, D. Posch, and H. Hellwagner, "Investigating the performance of pull-based dynamic adaptive streaming in NDN," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2130–2140, Aug. 2016.

[140] S. Petrangeli, N. Bouten, M. Claeys, and F. D. Turck, "Towards SVC-based adaptive streaming in information centric networks," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun. 2015, pp. 1–6.

[141] C. Xu, W. Quan, A. V. Vasilakos, H. Zhang, and G.-M. Muntean, "Information-centric cost-efficient optimization for multimedia content delivery in mobile vehicular networks," *Comput. Commun.*, vol. 99, pp. 93–106, Feb. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366416302729

[142] A. Detti, B. Ricci, and N. Blefari-Melazzi, "Mobile peer-to-peer video streaming over information-centric networks," *Comput. Netw.*, vol. 81, pp. 272–288, Apr. 2015, doi: 10.1016/j.comnet.2015.02.018.

[143] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[144] J. Yang, K. Zhu, Y. Ran, W. Cai, and E. Yang, "Joint admission control and routing via approximate dynamic programming for streaming video over software-defined networking," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 619–631, Mar. 2017.

[145] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using OpenFlow-assisted adaptive video streaming," in *Proc. ACM SIGCOMM Workshop Future Human Centric Multimedia Netw. (FhMN)*, 2013, pp. 15–20. [Online]. Available: http://doi.acm.org/10.1145/2491172.2491181

[146] A. Farshad, P. Georgopoulos, M. Broadbent, M. Mu, and N. Race, "Leveraging SDN to provide an in-network QoE measurement framework," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 239–244.

[147] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards QoE-aware video streaming using SDN," in *Proc. IEEE Glob. Commun. Conf.*, Austin, TX, USA, Dec. 2014, pp. 1317–1322.

[148] Q. Wang *et al.*, "GENI Cinema: An SDN-assisted scalable live video streaming service," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, Oct. 2014, pp. 529–532.

[149] S. Petrangeli, T. Wauters, R. Huysegems, T. Bostoen, and F. De Turck, "Software-defined network-based prioritization to avoid video freezes in HTTP adaptive streaming," *Int. J. Netw. Manag.*, vol. 26, no. 4, pp. 248–268, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.1931

[150] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering stable high-quality video: An SDN architecture with DASH assisting network elements," in *Proc. 7th Int. Conf. Multimedia Syst. (MMSys)*, 2016, p. 4. [Online]. Available: http://doi.acm.org/10.1145/2910017.2910599

[151] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP adaptive streaming using software defined networking," in *Proc. ACM Multimedia Conf. (MM)*, 2016, pp. 1296–1305. [Online]. Available: http://doi.acm.org/10.1145/2964284.2964332

[152] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "SDNHAS: An SDN-Enabled Architecture to Optimize QoE in HTTP Adaptive Streaming," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2136–2151, Oct. 2017.

[153] A. Bentaleb, A. C. Begen, and R. Zimmermann, "ORL-SDN: Online reinforcement learning for SDN-enabled HTTP adaptive streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 3, pp. 1–28, Aug. 2018. [Online]. Available: http://doi.acm.org/10.1145/3219752, doi: 10.1145/3219752.

[154] A. Bentaleb, A. C. Begen, and R. Zimmermann, "QoE-aware bandwidth broker for HTTP adaptive streaming flows in an SDN-enabled HFC network," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 575–589, Jun. 2018.

[155] C.-F. Lai, R.-H. Hwang, H.-C. Chao, M. M. Hassan, and A. Alamri, "A buffer-aware HTTP live streaming approach for SDN-enabled 5G wireless networks," *IEEE Netw.*, vol. 29, no. 1, pp. 49–55, Jan./Feb. 2015.

[156] A. Ganjam *et al.*, "C3: Internet-scale control plane for video quality optimization," in *Proc. 12th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Oakland, CA, USA, 2015, pp. 131–144. [Online]. Available: https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/ganjam

[157] J. Jiang *et al.*, "CFA: A practical prediction system for video QoE optimization," in *Proc. 13th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Santa Clara, CA, USA, 2016, pp. 137–150. [Online]. Available: https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/jiang

[158] Y. Sun *et al.*, "CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *Proc. ACM SIGCOMM Conf. (SIGCOMM)*, 2016, pp. 272–285. [Online]. Available: http://doi.acm.org/10.1145/2934872.2934898

[159] J. Jiang, S. Sun, V. Sekar, and H. Zhang, "Pytheas: Enabling data-driven quality of experience optimization using group-based exploration-exploitation," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, MA, USA, 2017, pp. 393–406. [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/jiang

[160] E. Thomas, M. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey, "Enhancing MPEG DASH performance via server and network assistance," in *Proc. Int. Broadcast. Conv. (IBC) Conf.*, Amsterdam, The Netherlands, 2015, pp. 48–53.

[161] J. W. Kleinrouweler, B. Meixner, and P. Cesar, "Improving video quality in crowded networks using a DANE," in *Proc. 27th Workshop Netw. Oper. Syst. Support Digit. Audio Video (NOSSDAV)*, 2017, pp. 73–78. [Online]. Available: http://doi.acm.org/10.1145/3083165.3083167

[162] J. Famaey *et al.*, "On the merits of SVC-based HTTP adaptive streaming," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, Ghent, Belgium, May 2013, pp. 419–426.

[163] Y. Sanchez *et al.*, "Efficient HTTP-based streaming using scalable video coding," *Signal Process. Image Commun.*, vol. 27, no. 4, pp. 329–342, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596511001147

[164] U. S. M. Dayananda and V. Swaminathan, "Investigating scalable high efficiency video coding for HTTP streaming," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun. 2015, pp. 1–6.

[165] C. Timmerer, "MPEG Column: 120th MPEG meeting in Macau, China," *SIGMultimedia Rec.*, vol. 9, no. 3, p. 4, Jan. 2018. [Online]. Available: http://doi.acm.org/10.1145/3178422.3178426

[166] M. Belshe and R. Peon. (2012). *SPDY Protocol*. [Online]. Available: https://www.chromium.org/spdy/spdy-whitepaper

[167] M. Belshe, M. Thomson, and R. Peon. (2015). *Hypertext Transfer Protocol Version 2 (HTTP/2)*. [Online]. Available: https://tools.ietf.org/html/rfc7540

[168] J. Iyengar and M. Thomson. (2017). *QUIC: A UDP-Based Multiplexed and Secure Transport*. Accessed: Dec. 18, 2017. [Online]. Available: https://tools.ietf.org/html/draft-ietf-quic-transport-08

[169] C. Mueller, S. Lederer, C. Timmerer, and H. Hellwagner, "Dynamic Adaptive Streaming over HTTP/2.0," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2013, pp. 1–6.

[170] C. Timmerer and A. Bertoni, "Advanced transport options for the dynamic adaptive streaming over HTTP," *CoRR*, vol. abs/1606.00264, 2016. [Online]. Available: http://arxiv.org/abs/1606.00264

[171] D. Bhat, A. Rizk, and M. Zink, "Not so QUIC: A performance study of DASH over QUIC," in *Proc. 27th Workshop Netw. Oper. Syst. Support Digit. Audio Video*, 2017, pp. 13–18. [Online]. Available: http://doi.acm.org/10.1145/3083165.3083175

[172] M. Xiao, V. Swaminathan, S. Wei, and S. Chen, "Evaluating and improving push based video streaming with HTTP/2," in *Proc. 26th Int. Workshop Netw. Oper. Syst. Support Digit. Audio Video (NOSSDAV)*, 2016, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/2910642.2910652

[173] W. Cherif, Y. Fablet, E. Nassor, J. Taquet, and Y. Fujimori, "DASH fast start using HTTP/2," in *Proc. 25th ACM Workshop Netw. Oper. Syst. Support Digit. Audio Video*, 2015, pp. 25–30. [Online]. Available: http://doi.acm.org/10.1145/2736084.2736088

[174] R. Huysegems *et al.*, "HTTP/2-based methods to improve the live experience of adaptive streaming," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 541–550. [Online]. Available: http://doi.acm.org/10.1145/2733373.2806264s

[175] G. Carlucci, L. De Cicco, and S. Mascolo, "HTTP over UDP: An experimental investigation of QUIC," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, 2015, pp. 609–614. [Online]. Available: http://doi.acm.org/10.1145/2695664.2695706

[176] R. Netravali *et al.*, "Mahimahi: Accurate record-and-replay for HTTP," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2015, pp. 417–429. [Online]. Available: https://www.usenix.org/conference/atc15/technical-session/presentation/netravali

[177] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7, doi: 10.1109/ICC.2017.7996611.

[178] R. Ghaznavi-Youvalari *et al.*, "Comparison of HEVC coding schemes for tile-based viewport-adaptive streaming of omnidirectional video," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–6.

[179] M. Graf, C. Timmerer, and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over HTTP: Design, implementation, and evaluation," in *Proc. 8th ACM Multimedia Syst. Conf.*, 2017, pp. 261–271. [Online]. Available: http://doi.acm.org/10.1145/3083187.3084016

[180] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "Improving virtual reality streaming using HTTP/2," in *Proc. 8th ACM Multimedia Syst. Conf.*, 2017, pp. 225–228. [Online]. Available: http://doi.acm.org/10.1145/3083187.3083224

[181] C. Timmerer, "Immersive media delivery: Overview of ongoing standardization activities," *IEEE Commun. Stand. Mag.*, vol. 1, no. 4, pp. 71–74, Dec. 2017.

[182] B. Choi, Y.-K. Wang, M. M. Hannuksela, Y. Lim, and A. Murtaza, *Information Technology—Coded Representation of Immersive Media (MPEG-I)—Part 2: Omnidirectional Media Format*, ISO/IEC Standard FDIS 23090-2, Dec. 2017.

[183] D. Podborski *et al.*, "Virtual reality and DASH," in *Proc. Int. Broadcast. Conv. (IBC) Conf.*, 2017, pp. 1–11.

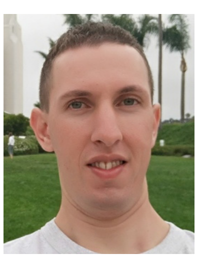
[184] S. Lederer *et al.*, "Distributed DASH dataset," in *Proc. 4th ACM Multimedia Syst. Conf. (MMSys)*, 2013, pp. 131–135. [Online]. Available: http://doi.acm.org/10.1145/2483977.2483994

[185] J. Le Feuvre, J.-M. Thiesse, M. Parmentier, M. Raulet, and C. Daguet, "Ultra high definition HEVC DASH data set," in *Proc. 5th ACM Multimedia Syst. Conf. (MMSys)*, 2014, pp. 7–12. [Online]. Available: http://doi.acm.org/10.1145/2557642.2563672

[186] J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Datasets for AVC (H.264) and HEVC (H.265) evaluation of dynamic adaptive streaming over HTTP (DASH)," in *Proc. 7th Int. Conf. Multimedia Syst. (MMSys)*, 2016, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/2910017.2910625

[187] A. Zabrovskiy, C. Feldmann, and C. Timmerer, "Multi-codec DASH dataset," in *Proc. 9th ACM Multimedia Syst. Conf. (MMSys)*, 2018, pp. 438–443. [Online]. Available: http://doi.acm.org/10.1145/3204949.3208140

[188] J. J. Quinlan and C. J. Sreenan, "Multi-profile ultra high definition (UHD) AVC and HEVC 4K DASH datasets," in *Proc. 9th ACM Multimedia Syst. Conf. (MMSys)*, 2018, pp. 375–380. [Online]. Available: http://doi.acm.org/10.1145/3204949.3208130



Abdelhak Bentaleb received the M.S. degree in computing (network and multimedia) from Mohamed El Bachir El Ibrahimi University, Bordj Bou Arréridj, Algeria, in 2011. He is currently pursuing the Ph.D. degree in computer science with the School of Computing, National University of Singapore, Singapore. His research interests include multimedia systems and communication, video streaming architectures, content delivery, distributed computing, computer networks and protocols, wireless communications, and mobile networks. Further information can be found at https://www.comp.nus.edu.sg/∼bentaleb/.

Bayan Taani received the B.Sc. degree in network engineering and security from the Jordan University of Science and Technology, Irbid, Jordan, in 2014. She is currently pursuing the Ph.D. degree in computer science with the National University of Singapore, Singapore. Her research interests include multimedia systems and communications, adaptive video streaming, and virtual reality.

Ali C. Begen (S'98–M'07–SM'12) received the Ph.D. degree in electrical and computer engineering from Georgia Tech. He joined the Computer Science Department, Ozyegin University, Turkey. He was a Research and Development Engineer with Cisco, where he designed and developed algorithms, protocols, products, and solutions in the service provider and enterprise video domains. In addition to teaching and research, he provides consulting services to industrial, legal, and academic institutions through Networked Media, a company he co-founded. He was a recipient of number of scholarly and industry awards. He has editorial positions in prestigious magazines and journals in the field. He is a Senior Member of ACM. In 2016, he was elected as a Distinguished Lecturer by the IEEE Communications Society. Further information on his projects, publications, talks, teaching, standards, and professional activities can be found at http://ali.begen.net.

Christian Timmerer (M'08–SM'16) is an Associate Professor with Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria. He is a Co-Founder of Bitmovin Inc., San Francsico, CA, USA, as well as the CIO and the Head of Research and Standardization. He has co-authored seven patents and over 190 publications in workshops, conferences, journals, and book chapters. He participated in several EC-funded projects, notably DANAE, ENTHRONE, P2P-Next, ALICANTE, SocialSensor, ICoSOLE, and the COST Action IC1003 QUALINET. He also participated in ISO/MPEG work for several years, notably in the areas of MPEG- 21, MPEG-M, MPEG-V, and MPEG-DASH. His research interests include immersive multimedia communications, streaming, adaptation, and quality of experience. He was the General Chair of WIAMIS 2008, QoMEX 2013, ACM MMSys 2016, and Packet Video 2018. Further information can be found at http://blog.timmerer.com.

Roger Zimmermann (M'93–SM'07) received the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, USA, in 1994 and 1998, respectively. He is currently an Associate Professor with the Department of Computer Science, National University of Singapore (NUS), Singapore, where he is also the Deputy Director with the Smart Systems Institute, and co-directed the Centre of Social Media Innovations for Communities. He has co-authored a book, seven patents, and over 200 conference publications, journal articles, and book chapters. His research interests include streaming media architectures, distributed systems, mobile and geo-referenced video management, collaborative environments, spatio-temporal information management, and mobile location-based services. He is a Distinguished Member of the ACM. Further information can be found at http://www.comp.nus.edu.sg/∼rogerz/.