A Contemporary Survey on Live Video Streaming from a Computation-Driven Perspective

NHU-NGOC DAO, Sejong University, South Korea
ANH-TIEN TRAN, Chung-Ang University, South Korea
NGO HOANG TU, Seoul National University of Science and Technology, South Korea and Ho Chi Minh City University of Transport, Vietnam
TRAN THIEN THANH, Ho Chi Minh City University of Transport, Vietnam
VO NGUYEN QUOC BAO, Posts and Telecommunications Institute of Technology, Vietnam
SUNGRAE CHO, Chung-Ang University, South Korea

Live video streaming services have experienced significant growth since the emergence of social networking paradigms in recent years. In this scenario, adaptive bitrate streaming communications transmitted on web protocols provide a convenient and cost-efficient facility to serve various multimedia platforms over the Internet. In these communication models, video content is delivered optimally, possibly transcoded, edited automatically, and cached temporarily by network elements along the path. To this end, the computational capabilities of various network elements are considered as major resources to be optimized for service quality improvements. This paper provides a contemporary survey of cutting-edge live video streaming studies from a computation-driven perspective. First, an overview of the global standards, system architectures, and streaming protocols is presented. Next, hierarchical computation-driven models of live video streaming are anatomized, including cloud-, edge-, and peer-to-peer-based solutions. Cutting-edge studies are then reviewed to discover the advances they have made in improving system performance in multiple aspects. Finally, open challenges are presented to direct future research in this field.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Information systems → Computing platforms**; **Multimedia streaming**; • **Networks → In-network processing**.

ACM Reference Format:
Nhu-Ngoc Dao, Anh-Tien Tran, Ngo Hoang Tu, Tran Thien Thanh, Vo Nguyen Quoc Bao, and Sungrae Cho. 2022. A Contemporary Survey on Live Video Streaming from a Computation-Driven Perspective. 1, 1 (February 2022), 37 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Authors' addresses: Nhu-Ngoc Dao, Sejong University, Department of Computer Science and Engineering, Seoul 05006, South Korea, nndao@sejong.ac.kr; Anh-Tien Tran, Chung-Ang University, School of Computer Science and Engineering, Seoul 06974, South Korea, attran@uclab.re.kr; Ngo Hoang Tu, Seoul National University of Science and Technology, Department of Smart Energy Systems, Seoul 01811, South Korea and Ho Chi Minh City University of Transport, Department of Computer Engineering, Ho Chi Minh City 710372, Vietnam, ngohoangtu@seoultech.ac.kr; Tran Thien Thanh, Ho Chi Minh City University of Transport, Department of Computer Engineering, Ho Chi Minh City 710372, Vietnam, thanh.tran@ut.edu.vn; Vo Nguyen Quoc Bao, Posts and Telecommunications Institute of Technology, Wireless Communications Department, Ho Chi Minh City 710372, Vietnam, baovnq@ptithcm.edu.vn; Sungrae Cho, Chung-Ang University, School of Computer Science and Engineering, Seoul 06974, South Korea, srcho@cau.ac.kr.

1 INTRODUCTION

The popularity of multimedia distribution platforms such as YouTube, Netflix, Twitch, and Facebook Live has led to an exponential increase in emerging social networking paradigms. In addition, recent user devices equipped with various computational capabilities and display resolutions adequately accommodate user satisfaction with video quality adaptation on demand. The technical advancements and convenience of multimedia delivery services has had a significant influence on market expansion [95]. For instance, personal live streaming content is simply produced to be posted on Facebook by any person who has basic knowledge of using digital devices. At home, smart TVs provide us with live entertainment content broadcast through various installable streaming apps released by content providers as well as third parties. In addition, online learning and meeting platforms such as Zoom, Cisco Webex, Google Meet, and Microsoft Teams are playing an essential role in supporting remote collaborations amid the Coronavirus disease of 2019 (Covid-19) pandemic by offering live video conferencing services. These analytical observations imply that live video streaming (LVS) services are expected to retain their dominance in Internet services in the following years [52].

From a technological perspective, LVS refers to a video delivery service that simultaneously records and broadcasts media content to all users in real time. To offer convenient service experiences, LVS is typically implemented on the Internet infrastructure using web transfer protocols to synchronously distribute video packets via multiple paths [181]. In a modern LVS system, heterogeneous user demands and preferences are supported by adaptive bitrate streaming (ABS) services that enable networks to dynamically adjust the quality level of the videos according to environmental conditions and resource availability. Network elements on the path play essential roles in optimally delivering, possibly transcoding, automatically editing, and temporarily caching video content during streaming operations. However, these network elements are typically limited by computation and storage resource constraints, which may prevent their efforts to perform these tasks. In addition, the instability and uncertainty of network conditions negatively affect the adaptation capability of the networks to dynamically adjust the video bitrate [34, 161]. Hence, although existing approaches have demonstrated several impressive advantages, LVS studies still attract considerable attention from both the research community and industry with the aim of improving the quality of experience (QoE), quality of service (QoS), and performance of LVS. Therefore, a contemporary review of cutting-edge studies on improving LVS performance is crucial to direct ongoing and future work in the field.

In the past, the literature has encompassed several surveys conducted on streaming services [16, 46, 179]. However, almost all existing studies have focused on streaming services by considering specific domains, such as delivery protocols, video applications, and performance metrics. In addition, they do not have taxonomized LVS and on-demand video streaming in the general streaming service category. Table 1 presents a summary of recent streaming service surveys. It is observed that the related work has its own limitations in responding to the aforementioned research questions in two ways: (*i*) the scope of existing studies covers streaming services in general instead of distinguishing the LVS, and (*ii*) in-network computation capabilities have not yet been considered as a major field of survey. As emerging edge-cloud computing paradigms have recently been integrated into every Internet service [33], these areas on which existing studies are rare have inspired us to conduct a contemporary survey on LVS from a computation-driven perspective.

To provide a comprehensive outlook of computation-driven LVS research, our survey was constructed as follows.

- First, we provide an overview of state-of-the-art commercial LVS platforms. We exploit the service qualities offered by various LVS providers, such as video dimension, maximum file size, maximum duration, total storage, and compatible formats. Our observations reveal emerging trends in LVS services. The details are provided in Section 2.

Table 1. Summary of contemporary surveys on streaming services.

Ref.	Research scope	Contributions	Year
[118]	Networking architecture	This paper provided a survey on information-centric networking architectures and models including content-centric and content delivery networks to support multimedia streaming applications.	2017
[147]	Bitrate adaptation	This paper reviewed client-side bitrate adaptation processes and protocols by considering multiple utility aspects. Pairs of optimization issues and feasible solutions for designing and deploying adaptation mechanisms were discussed.	2017
[104]	Bitrate adaptation	This paper investigated applicable techniques for client-side, server-side, and in-network bitrate adaptations to support video content delivery over the Internet. Video delivery architectures and ecosystems were elaborated on to illustrate appropriate techniques according to specific streaming scenarios.	2017
[182]	Access scheme	This paper described an overview of medium access algorithms in wireless local area networking environments for robust audio–video streaming services. Technical collaborations among supportive 802.11 standards were analyzed to evaluate their potential for streaming service implementation.	2018
[170]	Transmission topology	This paper presented a survey on mobile video distribution models including device-to-device transmission and wireless heterogeneous access networks. In addition, supportive techniques, such as name-based routing and in-network caching, were reviewed in such models.	2018
[14]	QoE management	This paper provided a survey on QoE management in adaptive streaming systems from three perspectives including networking infrastructure, in-network computing and caching services, and emerging applications.	2019
[16]	Bitrate adaptation	This paper updated the review in [104] by considering more recent studies within a similar survey structure. In addition, a comprehensive comparison among bitrate adaptation methods was conducted from the QoE and networking aspects.	2019
[179]	Streaming applications	This paper presented a specific survey on adaptive 360° video streaming solutions by considering viewport-independent, viewport-dependent, and tile-based approaches in unicast and multicast deliveries.	2020
[46]	Audience retention	This paper investigated the consumption of multimedia content on mobile devices using online streaming platforms. The effects of user preferences, Internet connectivity, and video quality on streaming services were analyzed.	2020
Ours	LVS	Our study aims to provide a contemporary survey on LVS from a computation-driven perspective. Our major distinguishing contributions include a focus on live video services and deliberation on their supportive system architecture, service models, and performance metrics.	Now

- Second, we provide an overview of LVS services. In particular, global recommendations and standards managed by international organizations are described. Adopting the standards, the LVS system architectures, along with their service components and functions, are clarified. We then present well-known streaming protocols integrated into LVS systems. The details are provided in Section 3.
- Third, hierarchical computation-driven LVS models are investigated that are further classified into cloud-based, edge-based, peer-to-peer (P2P)-based, and hybrid streaming categories. Here, the exploitation of relevant computing capabilities to assist LVS services at different locations on the video delivery paths is anatomized. The details are provided in Section 4.
- Fourth, to evaluate the improvements of cutting-edge LVS solutions, we divide these works into several groups by performance metrics such as service availability (SA), video bitrate, end-to-end (E2E) latency, network QoS/QoE, system serviceability, hit ratio, resource consumption, and security and privacy. The details are provided in Section 5.
- Fifth, from previous analytical observations, we present open challenges to drive ongoing and future research toward LVS advancements and popularity. The details are provided in Section 6.

The main contributions of this study are as follows. This survey provides a reference framework for interested readers, along with cutting-edge knowledge and studies regarding LVS services. From a computation-driven perspective, three technical areas constituting the LVS were systematically investigated, including standard architectures, computing-assisted models, and metrics of performance evaluations. Moreover, the lessons learned are summarized and discussed at the end of each section. In addition, open challenges are highlighted to support future research.

2 STATE-OF-THE-ART COMMERCIAL LVS PLATFORMS

In the emerging social networking era, along with the quantitative proliferation of live video providers, their high providability and utilization to meet various on-demand LVS services must be guaranteed simultaneously. It is well known that Facebook, YouTube, Tiktok, and Instagram account for the largest number of several billion monthly active users [64, 154], from which, among the shared videos, live video platforms occupied one out of five in total. LVS providers make their best efforts to supply the highest service providability in terms of the minimum/maximum dimension, aspect ratio, maximum file size/video duration, total file storage, and compatible video format. As prime examples, Facebook Live [113], YouTube Live [114], Tiktok [160], and Instagram [86] are free to use, and are sufficient for streaming as a hobby with the ability to provide application programming interface (API); their commercial perspective is mainly from advertisements.

In comparison, owing to cross-platform services, Facebook Live and Youtube Live can flexibly support various video resolutions, maximum file sizes, maximum live duration, codec formats, and search engines compared to those of Tiktok and Instagram. In the alternative, Tiktok and Instagram are the more convenient choices with portable characteristics for everywhere and everywhen purposes because of their mobile app-based platform. Although Facebook Live, YouTube Live, Tiktok, and Instagram are undoubtedly useful in non-commercial scenarios, they suffer from disadvantages such as limited privacy tools, no monetization options, non branding removal, and minimal professional features. As a result, professional users would prefer to utilize other commercial LVS providers with more features and functionalities. Besides that, Twitch [163] is also known as a commercial LVS provider. Twitch does not charge content providers to use their platform but allows them to make money off on the users within different level-subscriptions that are offered as Tier 1, Tier 2, and Tier 3.

In the commercial service category, the key features of Dacast service are that it is well-known to be smooth running and provides cloud-based video transcoding, unlimited concurrent viewers and live channels, LVS recording, mobile device support, various monetization options, third-party player integration, security, no advertisements, ABS support, real-time analytics, and global content data network (CDN) delivery [31]. Another example is IBM Cloud Video, the features of which are summarized as high LVS resolution, mobile device support, API, CDN, enterprise LVS, monetization capability, reliability platform, suitability for a large number of clients, and diverse functions for broadcasters. However, additional fees are required to stream above 720p resolution; moreover, there is poor detailed subtitle support [120]. A brief description of the features of Vimeo Livestream are that it includes high LVS resolution, API, ABS, content management service (CMS), privacy/monetization options, no advertisements, unlimited events and viewers, professional interface, and detailed analytics; however, there is less traffic support [115]. The comprehensive characteristics of BrightCove are summed up as ABS, CDN, security and monetization options, customer relationship management, custom video portals, and enterprise-level features [21]. Meanwhile, [121] showed that Wowza was able to provide performance monitoring, high LVS resolution, mobile user support, API, LVS recording, cloud management portal, robust security, and video looping. However, it does not provide multi-streaming, multi-cameras, monetization, and scheduling. Kaltura and JW Player both support ABS, CDN, API, monetization capability, and robust security [99, 136]. However, Kaltura can provide a highly customizable platform that is complex and unsuitable for new broadcasters, whereas JW Player is an easy-to-use platform. For Muvi, the described characteristics for service providability are CDN delivery; it has HTML5 video player, CMS and analytic tools, and transcoding. No coding is required, and there is website and apps support for mobile and television, monetization, and security options [132]. The drawbacks are a complex CMS platform and bad integration. Stream Shark benefits include providing global multi-CDN services, mobile

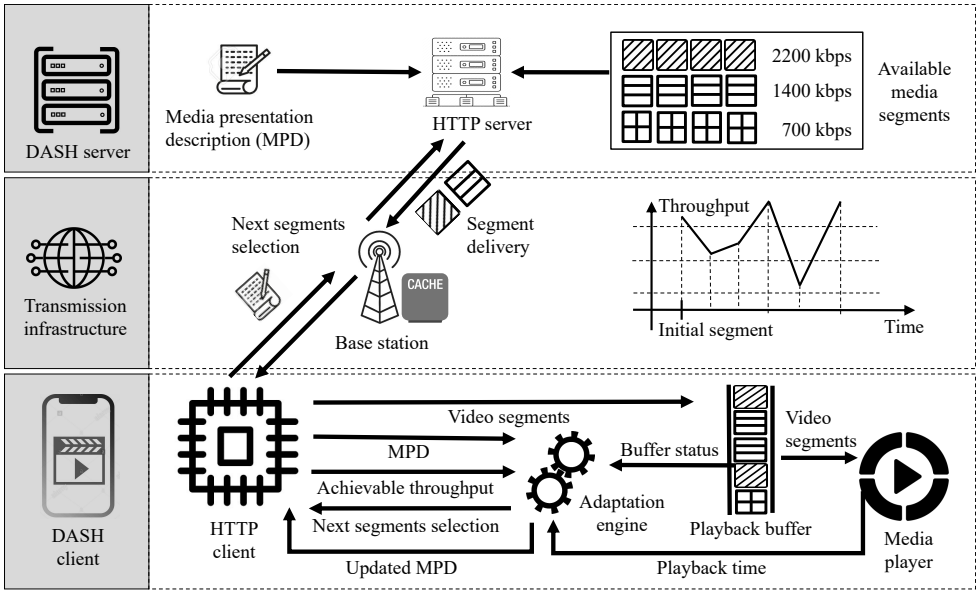


Fig. 1. Adaptive HTTP streaming in a DASH system.

compatibility, viewer reports, video encoding, monetization, and privacy options [155]. Note that video analytic and embeddable playlists are not included in Stream Shark's service providability. Finally, API for further integration, CMS, access and secure portal management, and analytic tools are simultaneously supported by Panopto despite the lack of customizable templates and an image editor [135].

3 LIVE VIDEO STREAMING SERVICES

LVS takes video digital signals and transfers them live online to multiple players all over the world. Starting at the camera, video frames are captured in real time and converted into continuous video signals. At 4K resolution, the signal can reach a bitrate of gigabits per second. A codec is used to compress the original video into smaller and manageable sizes. Codecs such as H.264 squeeze the bitrate from gigabits per second to megabits per second and then the data are packetized into transportation protocols (e.g., real-time messaging protocol (RTMP)) so that the videos can be streamed over the Internet. The video data are then transmitted to the media server, in which they are packetized into various communication protocols, such as HTTP live streaming (HLS), for delivery to multiple players, although the content of the stream remains unchanged. To further increase playback availability, content can be transcoded into new codecs, transrated into portable versions of various bitrates, and transformed into multiple versions of different resolutions. These processes enable LVS to support homogeneous devices with different resolutions and Internet connection speeds. As clients may be anywhere around the world, global CDN can be used for faster distribution and to reduce latency, making what is seen on the media player as close as possible to real life. From a system model perspective, this section presents a thorough discussion of recommendations, standards, and streaming protocols that have been developed for streaming services.

3.1 Recommendations and Standards

The Third Generation Partnership Project (3GPP) officially published adaptive HTTP streaming in 2009 [45]. The brief description of the media format is further elaborated in [44] in collaboration with the Moving Picture Experts Group (MPEG) working groups. The MPEG issued dynamic adaptive streaming over HTTP (DASH) standard in 2012 [88], and its latest version was published in 2019 [89]. Many companies have contributed to this standard, including Microsoft, Adobe, Apple, Samsung, Akamai, Dolby, Ericsson, Qualcomm, Netflix, Intel, and Bitmovin. The design principles of MPEG-DASH standard are to: (1) swiftly adapt to network condition fluctuations; (2) seamlessly switch video quality depending on sudden network impairments; (3) effortlessly reuse cache infrastructure; (4) easily bypass firewalls using HTTP messages; (5) actively provide high-quality user experience; (6) converge with existing proprietary technologies; (7) support all video and audio codecs and file formats; (8) conveniently deploy videos without additional hardware changes or updates; (9) proactively move intelligence from network to client; (10) enable advertisement insertion for commercial purposes.

To implement these design principles, MPEG-DASH enables clients to proactively and adaptively choose video quality during the entire viewing session. Each piece of video content, either from the video library or live events, is ingested by the media server to transcode it into more compatible versions and diversify the available representations to satisfy advanced viewers' setups. To effectively handle fluctuating channel conditions, DASH allows clients to automatically shift between deficient- and rich-video encodings in terms of resolution and bitrate. The renditions are conveyed through the HTTP protocol as a series of segments rather than a bulk file. The clients measure the current Internet connection speed and level of playback buffer to choose the next segments with the expectation that the next video segments are always available before the current video segment expires.

The DASH possesses its own shortcomings, especially in peak times, when multiple DASH clients must compete for shared network resources, such as bandwidth. Specifically, the research community has thoroughly investigated solutions for the problems of QoE unfairness among clients, the destructive influence of bitrate switching, screen freezes and initial delay, network resource under-utilization, outdated information in media presentation description (MPD) after a network failure or reconfiguration, or undesirable interactions and oscillations among DASH clients competing for the same bandwidth. These problems constitute a serious concern for video content providers and network operations and become worse in the case of diversified environments. To alleviate these problems, the server and network-assisted DASH (SAND), a finalized extension of the MPEG-DASH standard (in 2017), with the aim of enhancing the delivery of DASH content [90] was proposed. This will be discussed next. The SAND architecture has three broad categories of elements: *(i)* DASH clients, *(ii)* DASH-aware network elements, and *(iii)* regular network elements. Correspondingly, it requires three interfaces that bear diverse types of messages: *(i)* metrics and status (from clients to DANE), *(ii)* parameters enhancing delivery (PED) (among DANEs), and *(iii)* parameters enhancing reception (PER) (from DANE to clients). All these messages are referred to as SAND messages. SAND messages are not necessarily sent simultaneously.

Clients inform other elements of the network regarding their current status on the DANE via status messages. For instance, the client apprises the cache server whose specific segments are likely to download and, then, the cache server proactively prefetches them ahead and immediately serves segments as soon as the actual request from the client is sent. This process is expected to enhance the cache hit ratio on the server and the perceived QoE of clients. The cache server informs associating clients regarding available segments via PER messages. The DASH clients may consider these messages as a suggestion for the selection of future requests to retain a stable and continuous streaming

experience. Consider a live streaming scenario wherein a large number of DASH clients expect to watch the same content, for example, sports events/live concerts, and each DASH client possesses different capabilities in terms of network conditions. The QoE of clients can instantly deteriorate because the cache server cannot prefetch all requested segments owing to bandwidth and/or storage shortage. PER messages can help to lift this burden by notifying the clients of the available segments so that DASH clients can properly modify their requests. The server can communicate information regarding the streamed video to the network delivery element/node using a PED message. However, the SAND specification does not provide PED messages in the primary edition.

Besides, the video quality experts group (VQEG) [165] develops some tools called StreamSim [164] to simulate the streaming environment for research purposes. This toolchain can perform five tasks in a separate fashion including video encoding, streaming, loss insertion, payload extraction, and decoding. Each task is also possible to be individually configured and additional features could be considered. Specifically, the packet loss, delay, or jitter can be simulated via pre-defined network configurations, and the raw video material can be encoded with different settings and transmitted. The decoded transmitted videos would be used to compare with the original video.

3.2 System Architectures

Figure 1 describes a typical DASH session. The DASH server splits the original video into multiple video representations, each of which is available in differentiated quality levels and the corresponding MPD file. A segment is a unit of data associated with an HTTP-URL with a size specified by the MPD [71]. These media segments comply with the media format the system is associated with and enable playback either independently or when combined with other segments. Initially, the client sends an HTTP request to the DASH server and receives the corresponding MPD file. The main concerns at the server side are optimal encoding, choice of available representations, and segment length (where selectable). The selected segment length should satisfy two contradictory requirements. It should be long enough to maintain a low data overhead and short enough to quickly react to the oscillating network conditions. The segments can be cached for future requests as they traverse the base stations (BSs). The segments easily traverse through the firewalls using HTTP messages and then fill in the playback buffer, decode, and play by media players (such as THEOPlayer [158], Video.js [22], Flowplayer [1], Clappr [29], JWplayer [136], Bitmovin [84], and VLC media player [137]).

The viewing process will be interrupted if there are no remaining segments in the playback buffer, leading to degradation of the user's experience. To decrease the frequency and duration of stalling events (screen freezes owing to an empty playback buffer), the adaptation engine always updates technical parameters, including the buffer status, current playback time, and achievable throughput to properly determine the bitrate of the next video segment [148]. As shown in Fig. 1, even for the scenario of highly fluctuating network environments, the DASH client is expected to actively adapt the quality of future video segments. The throughput is initially sufficiently good to provide initialization segments with the highest quality (2200 kbps); it is then condensed to a lower level so that a reduced video quality may be served to avoid playback buffer emptiness (1400 kbps). Subsequently, the bandwidth is improved; it then abruptly decreases. All these abnormal changes in network throughput can be quickly observed at the DASH client side, and immediate response can be determined by the adaptation engine. In particular, if any reduction in bandwidth is detected, the DASH client may agree to downgrade video quality and size to prevent buffer emptiness and retain a seamless media consumption experience. In another case, if the bandwidth is enlarged, they can demand a higher visual quality, thereby achieving better QoE. The switching among different representations can be monitored during the playback because the segments corresponding to respective quality can be requested separately and then merged at the client side. The adaptation engine inside the DASH client updates the MPD file and sends it back to the DASH server.

As defined in the latest standard on adaptive video streaming, that is, IEEE 1857.7 [71], an MPD file is an extensible markup language (XML) document containing metadata for accessing segments and providing streaming media services for users. The metadata of the video segments include segment durations, video/audio codec, bitrate, and video spatial resolution. The format of the segment conforms to ISO/IEC 13818-1 [87], GB/T 20090.1 [50], or GB/T 17975.1 [49]. A complete MPD schema and details of MPD are presented in the IEEE 1857.7 standard [71]. Each MPD file consists of one or multiple periods (high-level time interval of the media presentation) and can be fragmented and partially delivered if sudden network impairments occur unexpectedly. The MPD can be updated proactively by clients during the streaming session. Periods determine the beginning and end times of each part of the media presentation and can be used to insert advertisements and content segments. Each period contains one or more adaptation sets, each of which is a set of compatible encoded versions of media presentations. Each adaptation set contains one or more perceptually equivalent representations and can construct media streams with the same media content components. Seamless switching across diverse representations was implemented by equipping the adaption set and its contained representations with sufficient information. The adaptation set also specifies the maximum and minimum bandwidths, widths, heights, and frame rates of their representations. Therefore, DASH can easily support a wide range of devices with different settings and capacities. Each representation is either a complete or a subset of media content components.

Representations can be encoded with different video codecs, allowing battery-powered devices to choose older codecs to reduce battery usage. The DASH clients might override the choices of quality of perceived video to satisfy their own preferences, such as willingness to have possible video stalls in exchange for higher quality or degradation of video quality for the sake of smoothness. The segments within a representation are optional for decoding or restoring representations. Moreover, if the segments are perfectly time-aligned, smooth switching can be achieved. Note that stream access points (SAPs) indicate the position in a representation from which clients can begin playback of a media stream utilizing solely the enclosed information in representation data initiating from that position onward.

3.3 Streaming Protocols

Traditional streaming protocols such as RTMP and real-time streaming protocol (RTSP)/real-time transport protocol (RTP) are the standard for transporting video over the Internet. RTMP [82] is a TCP-based protocol that was initially designed for audio/video and other data transmission between a streaming server and the Adobe Flash Player. It has multiple variations, including RTMP proper, RTMPS, RTMPE, RTMPT, and RTMFP. In RTMP, the client and server establish a connection by exchanging AMF [83] encoded messages. RTMP sessions are secured using either industry standard TLS/SSL mechanisms or RTMPE or RTMPS. Although RTMP utilizes only TCP, RTP runs over both UDP and TCP.

At least three big tech companies have commercially rolled out HTTP adaptive streaming solutions in parallel, namely, Adobe HTTP Dynamic Streaming (HDS) by Adobe Systems Inc. [3], HLS by Apple Inc [8], and Microsoft Smooth Streaming (MSS) by Microsoft Corporation [124]. These proprietary solutions are mutually incompatible and use different terminology and data formats, despite their similar technological backgrounds [148].

New open-source protocols such as secure reliable transport (SRT [56]) and Web real-time communications (WebRTC [39]) are expected to change the landscape with their target to reduce latency. WebRTC is a free project supported by Google, Mozilla, and Opera, among others. It aims to provide browsers and mobile applications with real-time communication capabilities (ultra low latency of 0.5 second) via simple APIs. (Compared to the Apple Common Media Application Format (CMAF) with the same purpose, CMAF provides a low latency standard of 3–5 seconds). SRT was developed and pioneered by Haivision to optimize streaming performance across fickle networks with secure streams

Table 2. Comparison of different adaptive streaming protocols.

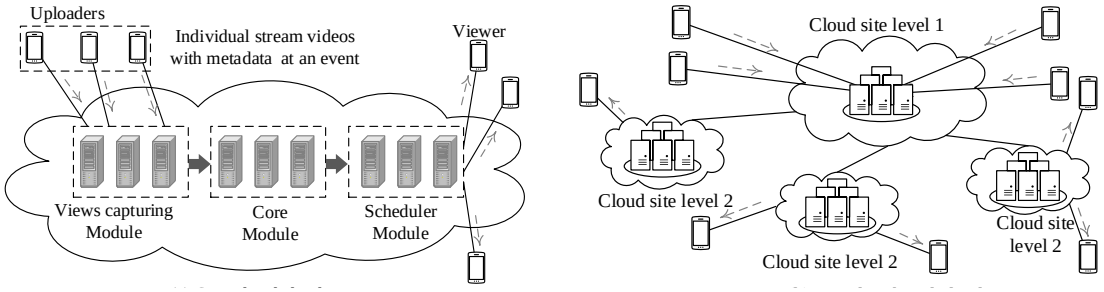
Protocols	Data description	Video codec	Audio codec	Playback compatibility	Encryption
MPEG-DASH [89]	MPD (XML)	Codec-agnostic	Codec-agnostic	All Android devices; most post-2012 Samsung, Philips, Panasonic, and Sony TVs; Chrome, Safari, and Firefox browsers	CENC [54], CBCS
HLS [80]	Playlist file (M3U8)	H.265, H.264	AAC-LC, HE-AAC+ v1 & v2, xHE-AAC, Apple Lossless, FLAC, MP3, AC3	All Google Chrome browsers; Android, Linux, Microsoft, and MacOS devices; several set-top boxes, smart TVs, and other players	CENC [54], CBCS, AES
MSS [126]	Manifest (XML)	H.264, VC-1	AAC, MP3, WMA	Microsoft and iOS devices, Xbox, many smart TVs, Silverlight player-enabled browsers	base64 encoding [72], CENC [54], PIFF [125], ASE
HDS [85]	Manifest (F4M)	H.264, VP6	AAC, MP3	Flash Player, Adobe AIR	Specific in use
RTMP [82]	Message (AMF [83])	H.264, VP8m VP6, V1, V2	AAC, MP3, SPEEX, OPUS, Vorbis	Flash Player, Adobe AIR, RTMP-compatible players	TLS, SSL, RTMPE, RTMPS
WebRTC [39]	Session Description Protocol (SDP) [75] (WebIDL [166])	H.264, H.265, VP8, VP9	Opus [78], PCMA & PCMU, DTMF [74] , iSAC, iLBC	Web browsers such as Chrome, Firefox, and Safari support WebRTC without any plugin	DTLS [76], SRTP [81]
RTSP/RTP [79]	SDP (UTF-8 [73])	H.264, VP9, VP8, MPEG-4 [77]	AAC, AAC-LC, HE-AAC+ v1 & v2, MP3, MPEG-4 [77], Speex, Opus, Vorbis	Quicktime Player and other RTSP/RTP-compliant players, VideoLAN VLC media player, 3GPP-compatible mobile devices	SRTP
SRT [56]	Data/Control Packet Header	Codec-agnostic	Codec-agnostic	VLC Media Player, FFPlay, Haivision Play Pro, Haivision Play, Larix Player, Brightcove, FFmpeg, OBS Studio, Libav	AES, OpenSSL

(empowered by AES [127]) and easy firewall traversal. Although any data type can be transferred via SRT, the protocol is particularly optimized for audio/video streaming. Haivision and Wowza founded a consortium (SRT Alliance) dedicated to the continued development and adoption of the protocol, and its current membership numbers more than 170.

The differences between the current streaming protocols are listed in Table 2. The term *codec-agnostic* means that the related protocol supports all codecs. Note that the common MPEG encryption schemes, CENC and CBCS, are mutually exclusive. Specifically, encrypted content according to CENC cannot be decrypted by a system supporting only the CBCS, and vice versa. The following properties are of high importance in the context of this survey: data description, video/audio codec, playback compatibility, and encryption.

3.4 Summary and Discussion

In this section, we have reviewed the recommendations and standards of LVS together with the detailed structure of DASH and presented a comparison table of commercial LVS protocols. In particular, Section 3.1 presents the design principles of MPEG-DASH, some challenges faced by DASH systems when employed in reality, and the DASH extension, SAND. Section 3.2 thoroughly describes the functionalities of the DASH system, the process of fetching content from the DASH server to the DASH client through transmission infrastructure, how clients modify their MPD file, and properly select the next video segments adapting to varying network conditions while keeping the viewing session smooth. To provide readers with a broader view, we describe the MPD file regarding the specification [71], from a hierarchical structure to a detailed semantic of its elements. Section 3.3 reviews the currently deployed LVS protocols. They can differ in data description, video and audio codec, playback compatibility, and encryption, but they share the same principle as generally aforementioned in Section 3.2.



(a) Centralized cloud. (b) Geo-distributed cloud.

Fig. 2. LVS streaming cloud-based models.

4 HIERARCHICAL COMPUTATION-DRIVEN LVS MODELS

4.1 Cloud-based Streaming

The mobile cloud computing (MCC) technique enables user devices to run computing services at a remote cloud data center via a high-rate and highly reliable air interface [36]. Some LVS services with scattered user characteristics, such as crowdsourced live streaming (CLS) or multi-party interactive live streaming [63] can be deployed by cloud-based platforms. To supply a more convenient LVS service, aiming at gaining users' satisfaction, large-scale cloud providers (e.g., Microsoft and Google) tend to deploy data centers globally forming geo-distributed clouds. As a result, findings in cloud-based LVS models primarily focus on how to provide smoothly enhanced LVS services in a one cloud site scenario, or how to deliver live videos effectively in geo-distributed cloud sites in terms of examining the use of resources, including computing, storage, and network resources.

4.1.1 Crowdsourced live streaming. CLS (i.e., crowdcast) services have emerged in the market, for example, streaming live sports online . Using multimedia distribution platforms for CLS, video contributors (including personal broadcasters and content service providers) can easily broadcast live streaming videos through their own devices. Video consumers located all over the world can watch live streaming videos in real time. Each LVS of each contributor, called single-source streaming, is aggregated to produce a crowdsourced streaming of a video channel. In CLS models, video contributors are generally called *crowdsourcers*, *broadcasters*, or *generators*; these include professional and non-professional uploaders. While sending live videos, crowdsourcers can chat and interact with their viewers through the system live chatting service. Evidently, CLS services demand efficient content collection, processing, and distribution with stringent delay constraints. Hence, when they meet a cloud system equipped with a powerful networking connection, all the mentioned challenges can be dealt with thoroughly.

A single centralized cloud-based system proposed in [19] can provide multiview LVS services to viewers by combining videos captured by multiple crowdsourcers watching an event (e.g., a game or concert). The difference in crowdsourcers' positions and viewpoints can supply viewers with the choice to select different views of the same real-time event. The proposed cloud-based multi-view crowdsourced streaming (CMVCS) system was designed as a modular concept; hence, it can be easily enhanced by adding more compatible modules to provide future integrated services. As shown in Fig. 2(a), three major modules with separated functions can be deployed in different hardware components. The *view-capturing* module plays a role in collecting raw videos as well as sending *metadata* (e.g., location, time, capturing

angle, name of the location, and captured event) that support multiple views. This module is installed in both the devices of crowdsourcers and viewers. Single-view LVSs are then sent to the *core module* for computing. The core module selects the best-quality video captured at the most appropriate angles among the received videos and classifies the video streams into different groups based on the captured angle. Each group provides a specific view for viewers. These selected views are then sent to the *scheduler module*, which performs transcoding by creating various representations. The best possible video representations will be sent to viewers based on the available bandwidth of the viewers and their selection. Both the core module and the scheduler module are located in the cloud, for example, cloud service providers' servers, that is, core module servers and transcoding servers. In these core module servers, an optimal algorithm is deployed to provide a joint solution for reducing the cost and maximizing QoE. In particular, the authors formulated a mixed integer programming optimization problem for resource allocation to create an optimal set of views and representations to maximize viewer satisfaction considering computational and communication resource constraints. In addition, the authors also proposed a heuristic algorithm called fairness-based representation selection (FBRS) to minimize the number of required transcoding servers. In particular, the set of views and their representations were chosen based on the average popularity of views. To prove the efficiency of the proposed system, the optimal, Top-N, and FBRS resource allocation algorithms were compared. Specifically, the QoE was investigated in terms of the computational instances and bandwidth. The results show that the proposed optimal algorithm yielded the highest values among the three algorithms.

A lower cost model for large-scale live video providers (e.g., Twitch.tv, YouTube Live) than that in [19] is a geo-distributed cloud infrastructure. In this paradigm, the CLS service can be deployed by multilevel cloud sites distributed across different global geographical locations. Each cloud site resides in a data center composed of interconnected and virtualized servers. The server resources will be provisioned for CLS, for example, computation resources for collective production and transcoding. As shown in Fig. 2(b), the single-source CLS of a crowdsourcer is uploaded to the highest-level cloud site (cloud site level 1). These servers at cloud site level 1 perform the function of source video collection and scheduling decision making. Based on a specific optimal algorithm, the CLS videos are forwarded to the allocated cloud instances in cloud level 2. Subsequently, the original source stream is transcoded into a target quality version, and then broadcast to viewers. In particular, Bilal *et al.* [20] presented a cost-effective QoE-driven video control plane to choose an appropriate transcoding location (cloud site) and video representations to minimize the overall system cost in terms of the viewer's available bandwidth, average latency between viewers and transcoding location including switching delay, required video quality, and resource availability per cloud site. There are two proposed algorithms in the case of optimal and a heuristic called greedy minimal cost (GMC). However, the GMC heuristic algorithm rarely achieves optimum streaming because it cannot adapt to changes in load or users' behaviors.

Although the system in [20] was focused on viewers' aspects, the work [41] provided a joint solution for reducing operational costs, including video transcoding cost, bandwidth cost, VM rental cost, and video distributions, for CLS providers in terms of data center selection fitting for both crowdsourcers and their viewers. In particular, an optimal online strategy based on the Lyapunov optimization framework was proposed for a geo-distributed cloud platform that can work cost-effectively while ensuring good QoE for users. The source data center, which is the data center selected for a crowdsourcer, the targeted data centers that are selected to deal with their viewer requests, and the interaction delay between them are considered as the input controls to build a specific video distribution path for each of the CLS services that the crowdsourcer is using at the same time. Moreover, this online algorithm can be executed in parallel to serve each crowdsourcer independently.

Applying machine learning (ML) or reinforcement learning (RL) to seek more precise solutions in resource allocation has recently become a popular trend in cloud-based CLS research [13, 60]. These works presented forecasting models applying ML to minimize the cost to the content providers while providing a maximum QoE level for users by solving the over-provisioning of resources. In particular, the model in [59] concentrated on assigning storage resources, whereas the model in [60] focused on computation resources for video transcoding. The total cost was considered including the total storage cost, total serving request cost, and total migration cost (i.e., the total cost of moving a video replica through cloud sites). Based on the metadata shared in [12], Haouari *et al.* [59, 60] set up an offline database in which each geo-distributed cloud site has a collection of near viewers for each incoming live video. To minimize both the start-up delay of the video transmission and the cost for the content provider, the storage resources are allocated as close as possible to the viewers. Much more complicated than the optimization problem in [59], the database in [60] with user collection is classified for each video bitrate representation. This database is used to make decisions to allocate optimal transcoding resources for each near user of a cloud site to minimize the overall system cost while maximizing the viewer's QoE. To proactively reserve the exact transcoding resources for incoming live videos, ML was adopted to build distributed time-series resource forecasting models. Simulations to evaluate the performance of the proposed system were examined, including the optimal cost and average latency in terms of renting hours. Specifically, five ML algorithms were applied to predictive models (i.e., long short-term memory, gated recurrent unit, convolutional neural network, multilayer perceptron, and XGboost).

Similarly, the work [13] applied RL to build an online and proactively predictive model, called reinforcement learning for online and proactive resource allocation (RL-OPRA), to address the minimum operational cost (i.e., rental cost, dispatching and migration cost, and serving cost) optimization. This model outputs a database of the popularity of live videos that are based on video features (e.g., broadcasters, category, creation time, and date) at different geo-located cloud sites. This predictive model is used to select the relevant data centers located in clouds while offering the best QoE for live streaming viewers by reducing perceived delays. In particular, the proposed RL-OPRA predictive model is deployed in a centralized master server that orchestrates resource allocation. The work also showed that the RL methods can give the same result as the optimal solution, and provide a better result than greedy decisions such as the GMC algorithm. Furthermore, the RL approach was utilized [13] to continue learning to adapt to any system fluctuation.

4.1.2 Multi-party interactive live streaming. Online video conferencing services have been widely deployed for virtual, face-to-face communication among separate parties, especially in the ongoing Covid-19 situation. The use of this kind of communication can also reduce travel expenditure for not only global companies but also individuals. Other applications for online multimedia conferencing services include distance learning, online video meeting and multimedia multiplayer online games. Unlike the CLS service, the LVSs in multimedia conferencing are two-way streams instead of one-way streams. An application user who is taking part in an online conference sends their LVS and receives the LVSs from all the other participants concurrently. Cloud computing solution development for providing multimedia conferencing services can be classified into three key directions based on each part of the service provision: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). Each solution development targets specific users and involved objects. In particular:

In cloud-based architecture, a harmonization among SaaS, PaaS, and IaaS is of importance for optimal operation throughout the whole network. To solve this problem, a joint PaaS and IaaS architecture as proposed in [152] along with novel APIs at PaaS and conferencing subtract at IaaS. This holistic architecture works efficiently allowing multiple conference application providers to share one conferencing service at SaaS with the same service characteristic either

audio or video. In addition, it also provides on-the-fly scaling of the running conference features under the required QoS. To this end, the memory and CPU resources are integrated into the total amount of allocated resources to fit the needs of all participants. To verify the performance of the proposed system, measurements based on system performance metrics (i.e., resource allocation, scale time, conference start time, and participant joining time) were conducted under both suboptimal and over-provisioned conditions. Moreover, this model can be used by multiple-level application providers, experts as well as non-experts. Furthermore, this model was investigated in terms of the efficient resource allocation solution for media handling services including video mixing, transcoding, and compressing by solving an integer linear problem and its heuristic in [153].

4.2 Edge-based Streaming

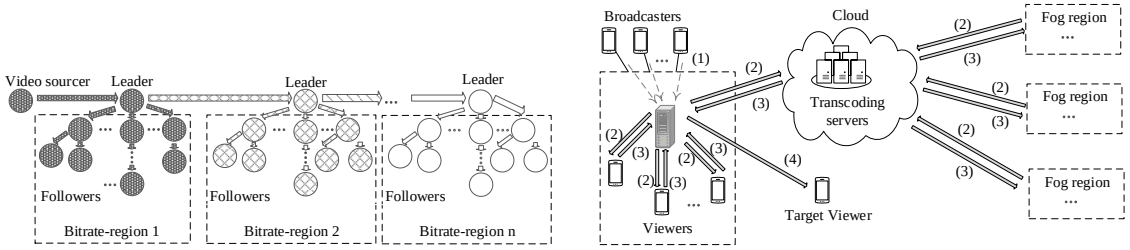
Multi-access edge computing (MEC), developed by the European Telecommunications Standards Institute (ETSI), offers cloud-computing capabilities for network management at network edges. In particular, MEC pushes the computing, storage, and control of network edges to the proximity of wireless users. Therefore, an MEC server can be a multi-task entity, including a streaming server to live streaming videos, a computation server for analytics or control, a video caching server, or a transcoding server. Significant studies focused on the functions of the MEC server in LVS models showed that MEC helps to enhance system performance.

A MEC-based routing algorithm [173] used for mobile users (i.e., participants) models the video streaming sent from a participant anticipating an online video conference as a multicast transmission process. In the system, each participant connects to the network via either wireless or wired links. The algorithm can flexibly run on any MEC server located at the network edge. A large number of MEC servers in the network function as forwarding nodes and continually transcode the video streams, as stated by the user request rates. The problem of constructing these multicast trees was modeled as a nonlinear integer programming problem with the aim of minimizing the total network resource cost and the MEC computing resource cost under multiple constraints, including satisfying the users' requirements for video rate and delay. The solution to this problem is solved using the proposed heuristic algorithm. To prove the effectiveness of this heuristic algorithm, the end-to-end delay and network resource utilization efficiency were compared with those of other multicast algorithms (i.e., SVC multicast algorithm and greedy multicast algorithm). The results showed that the proposed algorithm was the best among the compared algorithms.

Guo *et al.* [55] designed a joint video transcoding and video quality adaptation framework for ABS by utilizing a radio access network (RAN) with computing capability. Specifically, an ABS system in which video transcoding is performed at an MEC server in the vicinity of a RAN works under time-varying wireless channels. The MEC server is a combined entity that includes three components: the edge video server, transcoder server, and streaming server. The system model is configured to maximize the expected average reward, which is defined as the tradeoff between the cost of performing the MEC server's transcoding function and user-perceived QoE. Furthermore, the computational resource assignment and video quality adaptation are executed by applying an online automatic deep reinforcement learning (DRL) algorithm without knowing the channel information state. The proposed solution can significantly improve the ABS system performance compared with an ABS system without MEC. The disadvantage of this model is that it examines only slow-moving users, such as pedestrians.

4.3 P2P-based Streaming

Dogga *et al.* [40] proposed a P2P live video scheme in which both the transcoding and forwarding functions are performed at user devices (i.e., phone-based transcoding). The aim of this work was to maximize video liveness while



(a) Multibitrate-based transcoding and forwarding LVS P2P model. (b) Multi-fog paid transcoding LVS P2P model.

Fig. 3. Transcoding and forwarding LVS P2P-assisted models: (a) Voluntary peers [40], (b) Paid peers [192].

ensuring balance resource utilization at peers. The advantage of this scheme is that it is suitable for live video systems that fail to incorporate edge-based computation (e.g., congested edge-based transcoding servers). In addition, the requirement of bandwidth resources of fronthaul links connected between a peer and an equipment located at the network service provider (NSP; i.e., Internet service provider, cellular service provider) will be reduced. In [40], all the users watching the shared live video in the same bitrate are formed into a specific bitrate user collection (i.e., cluster), named the Bitrate-region, as shown in Fig. 3(a). The bitrate of a cluster obeys the rule: the higher the index of the Bitrate-region, the lower the bitrate. The participants (i.e., nodes) engaging in the system are categorized into three primary groups.

- *Video source*: the mobile user who uploads their live video after transcoding it to the highest requested bitrate.
- *A leader*: the mobile user who downloads the shared video content send by the video source or the next highest upper-level leader who forwards this video content to peers (i.e., followers) belonging to its Bitrate-region. In the case where a leader is not the leader belonging to the lowest Bitrate-region, it transcodes the video content to the next highest lower-level leader.
- *A follower*: the mobile user who downloads video shared by a leader or another follower.

Because a node enters or leaves a cluster at any time while the video service must remain available, the authors modeled each collection of viewers as a distributed balanced tree. To obtain the optimal solution for maximizing the liveness of the video service, a rebalanced algorithm is invoked locally in the cluster to balance device resources (i.e., bandwidth, energy) (*i*) when the number of nodes in this tree changes and (*ii*) to achieve fairness periodically. Clearly, the main disadvantage of the algorithm in [40] is that it does not provide an optimal solution for the entire D2D network. In addition, the number of hop-to-forward video content does not have an upper bound, which will lead to a significant end-to-end latency that may not meet some required QoE goals.

Another interesting research trend in LVS P2P systems is the formation of a cluster of peers that optimizes the cluster size. One sufficient P2P cluster in which peers view the same live channel can be formed into an alliance where only the contributing members are allowed to join. The existence of free-riders who benefit by cooperation with other users in D2D networks without contributing and redundant streams can drastically degrade playback quality and network performance. To reduce the influence of free-riders on a P2P live video system, Zhang *et al.* [184] presented a solution based on the distance-driven method for constructing a reciprocal P2P topology. Specifically, a group of truthful users (i.e., nodes) who contribute to and receive the assistance of other users in distributing data chunks within a group is formed gradually by the proposed distance-driven alliance algorithm. This algorithm can be invoked by the following

cases: (*i*) when a node joins the network and each peer is provided; (*ii*) when a node closer to the alliance becomes available during runtime. Working under these rules, the farthest member is replaced or a peer will be replaced if it does not show the contribution of chunks within the timeout period; this reduces redundant streams and shrinks the topology. Therefore, in contrast to earlier findings, this algorithm helps the D2D networks to operate efficiently in a large proportion of public IP nodes or in communication environments made vulnerable by traffic fluctuations. In addition, the performance of the proposed algorithm in terms of the continuity ratio (i.e., ratio of the received-before-played chunk count to the total requested chunk count) for different free-rider percentages is enhanced in comparison with other alliance algorithms including random alliance, bandwidth-likeness alliance, and content-likeness alliance.

4.4 Hybrid Solutions

In this section, we consider mainly cooperative models where cloud-assisted, edge-assisted, and peer-assisted entities help to improve the performance of LVS systems.

An adaptive bitrate control for low-delay multiparty interactive live streaming was presented in [171]. This model was designed to maximize the overall QoE of multiparty interactive live streaming. In general, a user device can be both a sender and a receiver. In [171], the authors assumed the user device to be a sender or receiver for each streaming section. The three logical entities in this system include user devices called senders, servers, and user devices called receivers. To reduce the computation in the cloud, which leads to an increase in the infrastructure cost, Wang *et al.* did not use the transcoding server located in the cloud. The transcoding function was assigned to the senders. Therefore, the sender sends a limited number of streams with bitrates required by their receivers to the server instead of a single stream, as in [41] via the uplink connection. In particular, each sender aggregates multi-rate streams into a video stream encoded using scalable video coding (SVC) and sends it to the server in frames. The server buffers the received frame and relays the relevant SVC layers of the frame to each receiver. An adaptive bitrate controller centrally located at the server site, named MultiLive, can provide a solution for personalizing the preferences between each pair of users. Specifically, MultiLive takes the inputs, including the uplink throughput of each sender, the downlink throughput of each receiver, and the state of the buffer occupancy at each receiver, to make decisions regarding (*i*) the set of bitrates that each sender should create and (*ii*) the bitrate that each receiver should receive. Interestingly, the average QoE score of this proposed system was shown to have the highest value in comparison with simplified algorithms, such as one-linear programming, buffer feedback adjustment, single, fixed, and Janus.

Based on goodput constraints at the application layer, the RAN analytic application for computing at the MEC server is effectively used to enhance the standard SVC for live downlink streaming videos. The encoded live videos from the video content server are transmitted to the MEC servers from the video content server located in the cloud. An edge computing algorithm deployed at MEC servers [186] located at BSs uses some input parameters, including the link status estimation and rate assignment, to calculate the optimized aggregation goodput performance of the input live video traffic. Instead of caching multiple versions of a video with different resolutions, MEC servers deploy the SVC result to achieve high video transmission efficiency. A D2D network that includes users watching the same LVS forms a helping network to assist other users by transiting the current video streaming content. In particular, users watching the same LVS are grouped into two types: those working under good channel quality and receiving LVS content directly from MEC servers, and weak users working under poor channel quality. These types of users can either receive direct packets from the serving MEC server or from other users under the assumption that the devices in the D2D network can access both cellular and D2D networks. In contrast to [40], the relaying hop in [186] is limited to one hop. This proposed model was evaluated through goodput (i.e., the amount of data received by the receiving end

within the effective time), average end-to-end delay, average effective loss rate, and average QoE, which is the relational expression of the former. The results showed that the proposed system is significantly superior to all measurement metrics in comparison with DASH-based method systems (e.g., MPEG-DASH, CDASH).

Another collaborative model of MEC servers and mobile users was proposed in [68]. This optimal QoS edge-based model transfers video content using in-network mobile computing located at BSs under budget constraints. Mobile user devices (i.e., generators) connected with a certain BS or an access point, via a platform supplied by a crowdsourced educational and entertaining application provider, upload their video content to upload servers (ULSs) attached to BSs. The ULSs with the video collector modules will forward the video contents of the received generators to appropriate download servers (DLSs) via backhaul links and the core network. DLSs then process these contents and distribute the processed data to viewers. To address the QoS of the video crowdsourcing platform, Huang *et al.* considered a group of generators cooperatively producing the same video content, which will then be forwarded to another group of interested users (i.e., viewers). Moreover, the system operates under budget constraints. The budget needed for content delivery from generators to viewers consists of two kinds of costs that the application providers must pay: (*i*) network data transmission cost, which is charged per byte, and (*ii*) server rental cost, which is assessed per unit time in both ULSs and DLSs. Choosing the optimal ULS and DLS for a given number of ULSs and DLSs for each generator and viewer in order to guarantee video crowdsourcing experiences under multi-level operational budget constraints is the problem to solve. To this end, a server placement and user association scheme was formulated as an optimization NP-hard problem. To verify the proposed system, the overall E2E delivery time reduction was investigated in terms of average video size per generator, the number of involved BSs, the number of users per collaboration group, and types of algorithms (i.e., brute force). In contrast to [192], three different practical budget cases classified into high, low, and medium levels were examined in [68]. Furthermore, the solution given in [41] was crowdsourcer-driven (i.e., multiple viewers are concerned about watching the content from one source), whereas the solution in [68] aims at the content delivery from multiple sourcers to multiple viewers. The limitation of this work is that the influence of immediate nodes (e.g., BS controllers and mobile switching centers) between the two selected ULS and DLS on the system performance was outside the scope of [68].

In [192], a CLS cloud-based system operated using viewers' phones with massive broadcasters. This peer-assisted model uses the idle end-viewers' resources to transcode immense video data to offload computational resources from the cloud. This solution reduces the leasing cost for content service providers and enhances the supply of low-latency LVS service stability. The system [192] operates in multiple regions with one regional data center (or a regional server) located in each region. The functions of this data center are as follows: (*i*) receiving the upload CLSs from broadcasters, (*ii*) assigning transcoding tasks to either viewers or cloud, and (*iii*) recollecting transcoded video and forwarding the processed streams for further delivery (Fig. 3(b)). An algorithm based on certain criteria, such as viewer stability, is used to select promising candidates who can assist the cloud and will be paid for their resource contribution (i.e., electrical power and computing). To deal with qualified viewer selection, the authors presented an auction-based approach that can be implemented in each region to concurrently implement two jobs: (*i*) enabling the crowd of viewers to facilitate the transcoding task assignments and (*ii*) offering a dynamic viewer-driven payment for these selected viewers under a given budget constraint. If the transcoding assignment cannot be deployed successfully (i.e., no satisfiable transcoding viewers can be chosen locally), the unmatched tasks will be directed to the cloud server. After processing the given transcoding task, the dedicated cloud server sends the transcoded stream back to the region. A prototype with an online scheduler was conducted to prove the feasibility of the design, and a comparison of three scheduler strategies (i.e., online, baseline, and comprehensive) in terms of the percentage of stable candidates, total cost, and total number of

reassignments was performed. Obviously, this model can be a valuable research direction because of its feasibility in utilizing idle resources from peers with payment. This policy contrasts with that in [40] and is suitable for constructing a long-lasting relationship between all involved entities in the network. This model can be considered for further improvements, as idle viewers can help concurrently with more than one job.

An edge-assisted crowdcast framework, called *DeepCast*, was proposed in [167]. For crowdcast content delivery, DeepCast seamlessly integrates many entities, including cloud, CDN, and non-uniform edge servers. In addition, through DRL, it automatically determines the most relevant strategies for viewer assignment and transcoding at edges. This proposed framework proved its effectiveness for better personalized QoE and lower cost for crowdcast systems. In this system, a broadcaster uploads their raw stream to a platform's service center (i.e., cloud). Next, the original stream is encoded and compressed into multiple-bitrate streams and pushed into the CDN servers. By using the WebRTC protocol or other proprietary protocols for multimedia streaming, service providers can provide interactive streaming services with a tight latency demand. The high-quality versions of streams from CDN servers are then forwarded to the edge servers through HTTP. These edge servers will possibly transcode the received data streams to low-quality versions in response to the different bitrate requests of viewers. To fulfill the joint requirements of minimizing the system cost and optimizing the viewers' personalized QoE, the regional edge can serve the viewer itself or ask for help from another edge or the CDN. To achieve low channel switching latency, the nearest of either of the two mentioned entities was chosen. To this end, the authors proposed a data-driven DRL-based approach located in an edge system that can automatically learn from the network and viewer information to make intelligent decisions without any predefined rules. Specifically, DeepCast applies the state-of-the-art asynchronous advantage actor–critic model [130] as the learning model. The three QoE metrics used in [167] are streaming delay, channel switching latency and bitrate mismatch level (i.e., a function of the difference between the target version of a viewer and the actual assigned version). Thus, the optimization objective is to minimize the sum of the overall penalty, including QoE and the system cost. Compared with other deep learning models, a deep Q-learning network (DQN) with its subcategories 1-step-DQN and n-step-DQN or Q-learning, the proposed system outperformed with regards to the overall penalty.

4.5 Summary and Discussion

In this section, we surveyed some cutting-edge application-specified (e.g., crowdsourced distance education, online conferencing, online interactive multiplayer games, crowdsourced entertainment) LSV models categorized into subsections based on the location where the computing is implemented. In these models, the devices can be heterogeneous phone cells or fixed devices (e.g., TVs, game consoles, or personal computers), and users can be professionals with cameras or amateurs. In addition, a range of service providers engaged in the systems are presented, including content service providers, network service providers, educational application providers, entertaining application providers, cloud service providers, edge service providers, conferencing service providers, and content owners. Moreover, ML techniques were applied to obtain more accurate results and system adaptation. From Table 3, these survey models are summarized based on grouped models, objectives, and constraints, where the column entitled "Multi" indicates whether the proposed model belongs to a multi-party interactive live streaming or not. In addition, we included our comments on some limitations that can be used in future driven-topic research in the scope of real-time LVS services.

Table 3. Related representatives of state-of-the-art hierarchical computation models.

Ref.	Multi	Hierarchical Models			Objectives	Constraints
		Cloud	Edge	P2P		
[19]		✓			Optimal set of representations	Computing and bandwidth resources
[20]		✓			Optimal local cloud sites at viewers	Required QoE (bandwidth consumption, computation limits, cloud site number, bitrate, delay, video quality)
[41]		✓			Optimal local cloud sites at crowdsourcers and viewers for minimized operational cost	Bitrate, data center choosing rule, threshold for average interaction delay
[59]		✓			Optimal cloud sites for storage at viewers that maximize the QoE of viewers and minimize total cost	Cloud site number, average serving request delay
[60]		✓			Optimal transcoding resource allocation and direct cloud site for each viewer	Cloud site location for specific tasks, bitrate, average serving-request delay
[13]		✓			Optimal and environmental-adaptable transcoding resource allocation in the viewers' proximity and the system's perceived delay reduction	Cloud site location for specific tasks, bitrate, average delay, average serving time
[152]	✓	✓			Maximized concurrent using system's user	Computing resource (i.e., CPU, memory)
[173]	✓		✓		Minimized cost (bandwidth resource cost and computing cost)	Video rate delay
[55]			✓		Maximized average reward	Computational (video transcoding) resource at the MEC server, channel condition, and playback buffer
[40]				✓	Maximized liveness of video service	Resources (bandwidth and energy), fairness, control overhead
[184]	✓			✓	Redundant stream reduction to form an optimal topology	Inter-user constraints
[171]	✓		✓	✓	Overall QoE maximization (i.e., delay, smoothness, quality, and stall)	Network uplink resource, buffer occupancy delay
[186]		✓	✓	✓	Maximized system's goodput	Cellular bandwidth resource, time transmission deadline
[68]	✓		✓	✓	Optimal MEC servers locations to maximize time consumed for all viewers	Operational budget
[192]	✓	✓	✓	✓	An auction-based approach to assign transcoding task and payment	Budget for paying idle helping peers, number job assignment
[167]	✓	✓	✓		DRL-based edge-assisted interactive crowdcast framework with personalized QoE	Edges' computation and bandwidth resources

5 PERFORMANCE METRICS

5.1 Service Availability

According to ITU-T E.860 [91] and X.140 [92] recommendations released by the International Telecommunication Union (ITU), SA refers to the probability that the system can work overtime to provide services with its satisfaction to users, whenever and wherever the services are required. In the context of LVS systems, SA metrics are alternatively measured by stalling duration over the total playback periods. For example, Dantas *et al.* investigated video on-demand streaming services, promising to easily extend LVS services by further considering the E2E latency, hosted in the cloud computing environment [32]; here the hierarchical modeling techniques used the Markov chains to deal with the complexity of representing such a system that focuses on the virtual machine and specific application components (e.g., web server and database server) required for video playback. In [32], the performance was achieved with an SA of 0.9881, which indicates a downtime of 104.24 hours per year. Meanwhile, Bezerra *et al.* [17] conducted an experiment to analyze the Eucalyptus platform for a video on-demand streaming system under cloud computing support, where (*i*) Eucalyptus is an open-source cloud middleware that is beneficial to the private cloud platform and (*ii*) the continuous-time Markov chain with reliability block diagrams was utilized to evaluate the SA metric as well as potentially demonstrate the extensive capability of LVS. The numerical results in [17] showed an SA of 0.988571 with an unavailability of 100.11 hours per year. To achieve a higher SA for the LVS service, in [123], Melo *et al.* proposed a redundant node architecture, where the secondary node controller (NC) has the same software and hardware specifications as the primary NC, which is only active when the primary NC fails. The results in [123] confirmed that the achievable SA and annual downtime

with the redundant node architecture were 0.990434 and 83.798 hours, respectively. By extending the work [123], Melo *et al.* further investigated the Eucalyptus cloud platform along with the design of experiments and percentage difference utilization to identify availability bottlenecks. Numerical results in [122] revealed that the value of SA is derived up to 0.994401; moreover, they revealed that the downtime degradation reached only 49.05 hours per year, which represents 2.04375 days of downtime in a year. However, the redundant node architecture's utilization has some tradeoffs among SA achievement, downtime, cost, and computational/employable complexity compared with the conventional approach. Furthermore, in [9], by additionally considering the occurrence of software aging issues in a web browser plug-in for cloud-based LVS services via two rejuvenation strategies, substantial performance improvement was achieved, including (*i*) the time-based rejuvenation strategy with 0.9999359 of SA representing 0.561516 h (33.69 min) of annual downtime and (*ii*) the prediction-based rejuvenation strategy with 0.9999361 of SA representing 0.559764 h (33.59 min) of downtime per year. In the proposed framework, the continuous-time Markov chain was leveraged to predict the resource utilization ahead of time, whereas an automated workload simulated the access behaviors of YouTube users.

5.2 Video Bitrate

The video bitrate is calculated by the amount of video data transferred in a given time unit, which is measured in bits per second [24]. A higher video bitrate is expected to achieve a higher video quality for the LVS experience. Owing to the unstable characteristics of many parameters affected by video quality, many authors have attempted to adopt the ABS concept [5, 57, 102, 149] to guarantee video quality as high as possible by adapting to bandwidth/throughput fluctuations owing to changes in network condition. In the ABS method, a transmitted video is simultaneously encoded at various levels of bitrates, from which these streams are divided into multiple segments and stored on an HTTP server, and the client will be assigned appropriate bitrate segments considering the network conditions [16, 104]. In particular, Han *et al.* proposed a cooperative client server based on HTTP adaptive streaming to provide a high-quality LVS service by improving bandwidth utilization [57]. The proposed model designs system operations as (*i*) the server adaptively encodes a live transmitted video into multiple video segments and (*ii*) the client chooses an appropriate segment bitrate by taking into account the quality, bandwidth, and buffered playback duration. Preliminary results in [57] showed that the LVS system can achieve higher bandwidth utilization and lower levels of adaptive bitrates than existing schemes. In addition, in [149], CDN has been considered as another key enabler to improve the quality of content delivered by LVS services, where the CDN infrastructure is responsible for providing fast delivery of Internet content via a geographically distributed group of servers that work together and the HLS protocol is further performed for LVS content. According to the results in [149], the LVS broadcaster achieved an 11.58% improvement in the average throughput (i.e., average video bitrate) and a 0.25% degradation in the average packet loss ratio, compared with a system without using CDN. Specifically, in the CDN incorporation with ABS mechanisms, Shafiq *et al.* presented an ABS measurement for a large-scale LVS event used by the CDN [5]. In the proposed framework, (*i*) clients estimated the network bandwidth and requested the appropriate bitrates and (*ii*) the Hampel filter for robust and efficient filter detection was utilized to detect spikes in residual subspace projections in real time, thus facilitating alarming system applications and monitoring video QoE impairments in real time. The numerical results in [5] have revealed that the Hampel filters can achieve 92% accuracy of the QoE impairment, which introduces an approximate 20% improvement in the true positive rate compared with baseline methodologies. In the same context, [102] considered a large-scale prototype with up to 500 LVS from users dynamically adapted at various bitrates under CDN integration. The system yielded an average user throughput increase of up to 70% compared with conventional benchmarks. However, the aforementioned mechanisms are mostly focused on high-quality LVS on best-effort networks, where a latency of several

tens of seconds may be exhibited, which is one of the foremost problems in LVS services. Section 5.3 is devoted to investigating effective solutions to fill this gap, where the ABS mechanisms are integrated with various strategies such as HTTP/2, DRL, video coding standards, cloud/edge/fog computing, etc.

5.3 End-to-End Latency

In LVS systems, E2E latency is considered a stringent constraint that defines the duration of the data transmission needed to traverse from a video source to playback clients [2]. The E2E latency consists of main delay factors (e.g., holding time and transmission delay) and occasional delays (e.g., propagation, radio access, queuing, and reordering delay) [4]. Among these factors, (*i*) the holding time characterizes the duration time needed to process or handle video frames on both the transmit and receive sides, (*ii*) the transmission and radio access delay refers to the duration of physical radio interface hardware to map the data from packets to bits, (*iii*) the propagation delay comes from the distance between terminals, (*iv*) the queuing delay refers to packet buffering at the terminals during transmission, and (*v*) reordering delay is caused by LVS on multipath networks. Consequently, to realize LVS services, the objective function is to minimize the E2E delay under the constraints of a given on-demand video quality, which has recently received more attention from scientists around the world.

Several studies have been conducted from a system model perspective to mitigate the E2E delay to facilitate LVS systems [69, 105, 108, 151, 156, 171]. For instance, Li *et al.* adopted the HTTP/2-based LVS framework to achieve low latency in video streaming, which was solved by the model predictive control frame-dropping algorithm [108]. The results in [108] indicated that the ABS method not only improves the achievable video quality and smoothness, but also reduces the frame size by 8.06% leading to significant E2E latency degradation. Similarly, the work in [105] was also presented, where two HTTP/2 features, including server push and stream termination, were leveraged in the LVS experience to enable low delay from the packet buffering that was minimized to 2 seconds. In particular, Shuai and Herfet [151] analyzed and obtained a closed-form expression for the average achievable buffering delay, that is, queuing delay, using the ABS method in the LVS system. Subsequently, Wang *et al.* [171] developed the MultiLive ABS algorithm for LVS services, where the E2E latency was reduced to approximately 100 ms. Furthermore, a novel DRL approach was recently developed in the low-latency viewpoint for LVS services. The work in [156] developed an ABS algorithm based on DRL, called DNNStream, which estimated the optimal video bitrate in the LVS experience for ultra-low-latency purposes. Meanwhile, in [69], the quality-aware rate control (QARC) algorithm based on DRL was proposed for LVS, which not only obtained an 18–25% improvement in the average video quality but also decreased 23–45% average E2E latency compared with Google Hangout [53], Compound TCP [138], and TCP Vegas [157].

From a transcoding perspective, typical publications that applied ABS based on video coding standards have significantly reduced the E2E latency for LVS services [98, 103, 142, 145]. As the first attempt, in [98], by leveraging the concept of ABS using the SVC for LVS services, the bitrate was controlled more frequently, resulting in coding bitrate decrements of 38% and a reduction in the E2E latency. Meanwhile, Kobayashi *et al.* [103] considered the ABS algorithm for LVS experience based on high-efficiency video coding (HEVC), also known as the H.265 video codec, which provides approximately double encoding efficiency compared with SVC, that is, 56.7% of the encoding bitrate improvement. Subsequently, Ryu *et al.* proposed an extension of HEVC, referred to as scalable HEVC (SHVC), which is applied for ultra-high-definition LVS [145] with scalability support, which showed a gain of approximately 20% decoding speed up. Furthermore, versatile video coding (VVC) is also a potential approach that provides a super video resolution up to 8K (7680×4320); it also conforms to the constraints of LVS applications [142], which is suitable for a richer user

experience of LVS services. In [142], their proposal provided a low initial queuing delay of approximately 0.21 seconds, which is 10 times lower than that of HTTP/2 in [105].

Utilizing in-network computing capability, with a focus on low-latency purposes, Bilal and Erbad [18] attempted to employ edge computing for interactive media and video streaming, where the latency and response time were minimized while providing outperformance of computing/bandwidth/energy savings in multimedia applications, transcoding, and video streaming. Similarly, Yang *et al.* [177] introduced an end–edge–cloud coordination framework to process LVS frames from different sources by considering the low-latency constraint as well as the accurate LVS analytic, LVS quality, and computing resource configuration. In [10], the fog architecture was highlighted by the effectiveness of not only low-latency but ultra-reliable communications for intelligent transport and video-on-demand scenarios. Meanwhile, the MEC paradigm was leveraged along with the flexible transcoding ABS to provide viewers with low-latency video-on-demand streaming services under the limited consideration of computing, caching, and bandwidth resources [110]. The experimental results from [110] have shown that the E2E latency is within the low range of 15–75 ms. It is worth noting that contemporary contributions [10, 110] are promising for extension of LVS services.

5.4 Network QoS/QoE

QoS measures key network performance metrics that focus on network characteristics such as jitter, latency, packet loss, rating factor, mean opinion score (MOS), etc., which do not take into account the relationship between the technology and the end-user. Meanwhile, QoE focuses on the actual individual user experience, which indicates whether the network actually delivers a sufficient end-user experience or not [14]. Some video-specific metrics can be utilized to quantify QoE that are widely accepted as good representations for end-users perceived quality in LVS systems, including the rate of buffering (RoB), buffering percentage over video session (BPoVS), rate of fluctuation (RoF), average playback bitrate (APB), frame video quality (FVQ), bitrate switch (BRS), frame skipping, resolution, latency, spectrum, MOS, etc. In [48], an adaptive SDN-based architecture with cloud mobile media was proposed, in particular for LVS applications, where the QoE was evaluated via the MOS metric. A factor analysis-based statistical method and a novel scheduling algorithm for an SDN controller were utilized to perform MOS estimation and SDN scheduling, respectively. Based on the evaluated results, the proposed methodology in [48] provided the end-user QoE improvement with high-accuracy MOS estimation compared to benchmarks (e.g., algorithms based on the spearman rank-order correlation coefficient, outlier ratio, and root-mean-square-error). Ahmed *et al.* [5] investigated the QoE in terms of RoB, BPoVS, RoF, and APB for a large-scale LVS event in the United States, where an ABS algorithm using server-side logs from a commercial CDN was employed to perform the LVS delivery for hundreds of thousands of viewers in the event. For the detection of QoE impairments, a principle component analysis-based technique and Hampel filters were designed, which offered 92% accuracy with a 20% improvement in the true positive rate as compared with baselines. Ren *et al.* [143] designed a greedy variable bitrate (GVBR) algorithm that optimized the QoE by simultaneously integrating three frameworks: (*i*) an appropriate key-frame interval that traded cross-frame compression for lowered inter-frame interdependency, (*ii*) a simple-yet-efficient frame dropping strategy to prevent excessive frame drops, and (*iii*) a bitrate adaptation strategy customized for broadcasters having shallow buffers. The proposed GVBR methodology in [143] not only achieved a comparable bitrate but also cut video interruption incidents by up to 90% compared to state-of-the-art algorithms. Subsequently, [51] introduced an edge-based transient holding of live segment (ETHLE) algorithm to tackle the high requirement problem of 4K-resolution LVS. According to the numerical results in [51], the QoE of the proposed LVS system in terms of initial startup delay, RoB, and latency was assured. In addition, the conventional transport-layer bottleneck was also addressed by utilizing virtualized caching resources at the mobile edge while guaranteeing high

data rate requirements. Yun *et al.* [180] proposed a QoE-driven resource allocation mechanism for LVS services of a cross-layer D2D link control system in D2D-underlaid fifth-generation cellular networks. The superior performance in terms of the QoE improvement, the average mean time to failure, the average peak signal-to-noise ratio (PSNR), and the average energy consumption of the proposed framework [180] has been demonstrated via system-level simulations. Meanwhile, Liu *et al.* [111] established a QoE-driven HTTP adaptive LVS channel placement (HASCP) strategy to optimize the channel allocation in media cloud servers, which led to QoE maximization and achieved higher bandwidth utilization than those based on benchmark solutions. In [25], a joint optimization problem of caching placement, video quality decision, and user association in LVS services under the dual pricing specification constraint was solved by a convex transformation and a one-step Lagrangian dual pricing algorithm. The proposed algorithm [25] achieved a remarkable enhancement of the average QoE per user in MEC-enabled cellular networks. Moreover, [37] designed a hybrid named data networking-based and Internet protocol-based (NDN-IP) prototype via operating system and networking virtualization techniques for LVS services to perform the efficient utilization of network resources and achieve a better QoE metric in terms of APB, BRS, RoB, and spectrum than conventional baselines.

Recently, ML-based applications have become more powerful artificial intelligence tools to effectively predict outcomes, in particular for network QoS/QoE measurements, without being explicitly programmed to do so. Specifically, Tian *et al.* [159] accelerated the training process of DRL-based QoE maximization via window completion with historical data and quick-start with a rate-based algorithm, named Deeplive, for LVS systems, where QoE measurements were taken into account in terms of RoB, FVQ, BRS, frame skipping, and latency. According to the experiment results in [159], Deeplive achieved not only low execution training time but also an average of $15-55\%$ improvement of QoE compared to state-of-the-art ABS LVS algorithms. [189] studied the user scheduling, transcoding decisions, and computational and wireless spectrum resource allocation problems in SDN-based cloud-aided heterogeneous networks, where the QoE function that was formulated as a logarithmic form was maximized under the constraint of a time-delay requirement. To tackle the problem of dynamic characteristics of wireless networks and the available resources with multi-dimensional continuous-discrete mixed variables, a Markov decision model with an online actor-critic learning algorithm was designed, which demonstrated its superior performance compared to the policy gradient algorithm and deep Q-learning network. In [116], an ML-based algorithm, namely ReCLive, was developed to effectively distinguish live streams from video-on-demand streams using media-request patterns as well as to infer QoE measurements in terms of resolution and RoB for the detected-chunk-attribute LVS. Furthermore, [30] introduced an innovative ML-based scheduling solution for omnidirectional LVS systems in highly dynamic unmanned aerial vehicle (UAV)-based environments. Based on the simulation results, the proposed methodology [30] has confirmed its effectiveness in terms of QoS provisioning, packet loss rate, PSNR, and throughput compared to state-of-the-art scheduling benchmarks (e.g., static prioritization, required activity detection scheduler, and frame-level scheduler).

5.5 System Serviceability

Although ABS algorithms have significantly gained multifold benefits for LVS systems as aforementioned investigation, unexpected outcomes of system serviceability may be considerably addressed, in addition to service instability, unfairness, and inefficiency. To analyze these issues, we consider a typical scenario in which LVS systems are constrained by limited resource capacities in terms of communication, computation, and storage. Typically, the service stability, fairness, and efficiency metrics are formally defined as follows [179]:

- *Service stability:* Service stability takes into account essential bitrate switches during LVS experience with the best effort to avoid unnecessary bitrate switches. This metric implies a video smoothness evaluation of LVS services.
- *Fairness:* Fairness metric represents the ability of the LVS systems to balance system resources to support multiple playback clients equally, which is modeled via Jain's fairness index [94].
- *Efficiency:* The efficiency metric exposes how efficient resource utilization was performed for the LVS services. Typically, the efficiency metric is measured by the number of video bits delivered successfully using a unit of resources such as bits-per-Hertz for bandwidth occupation and bits-per-Joule for energy consumption.

For instance, Jiang *et al.* proposed ABS-based fair, efficient, and stable adaptive (FESTIVE) for sharing a bottleneck link of multi-streaming in [97], where its performance was demonstrated to improve the service stability by 50%, fairness by 40%, and efficiency by 10% compared with various real and competitive commercial players. In [109], Li *et al.* innovatively proposed ABS-based PANDA, from which PANDA was able to improve the service stability by 75% and was significantly better in terms of fairness and efficiency than the conventional algorithms. However, there are tradeoffs between the service stability, efficiency, and fairness of PANDA when compared with FESTIVE. Meanwhile, an ABS proposal of the work [93] was implemented to provide improved stability, efficiency, and fairness metrics than the conventional one, where the authors utilize a logarithmic approach for received bandwidth that is increased or decreased logarithmically to converge to the fair share bandwidth, that is, the estimated bandwidth. In addition, in [133], Shahid Nabi *et al.* proposed a dynamic rate-adaptation algorithm, named SHANZ, to provide a balance between the service stability and efficiency even in drastic network fluctuations, where SHANZ was measured based on the adaptive step up function and feedback control mechanism. The results of [133] confirmed that the proposed method can achieve better balancing performance in terms of service stability and efficiency, compared with FESTIVE, PANDA, and another benchmark (e.g., the adaptation algorithm for adaptive streaming over HTTP, shortened by AAASH [128]). To further improve the performance of both the FESTIVE and PANDA strategies with respect to the stability, efficiency, and fairness metrics, [42] and [191] presented enhanced server and client cooperation (ESTC) and throughput-friendly DASH (TFDASH) novelty algorithms. In particular, in [42], ESTC allows fast convergence among different clients' bandwidth levels to the estimated bandwidth and establishes incorporation between the server and client sides to appropriately assign the allocated bitrate, where (*i*) the client has the responsibility for taking the right bitrate decision for the efficiency and service stability insurance, whereas (*ii*) the number of connected clients, current download bitrates, and bottleneck link bandwidth are leveraged at the server side to ensure fairness among competing clients. Meanwhile, the key idea behind TFDASH in [191] is to avoid the OFF periods during the downloading process for all clients by adopting a dual-threshold buffer model, for example, the low and high thresholds for preventing buffer underflow and overflow, respectively, to achieve a good balance among system serviceability factors.

5.6 Hit Ratio

In LVS networks, the hit ratio metric, that is, the caching hit ratio, refers to the server's capability to distribute the video streams that serve the most browsing users. The hit ratio is determined as the ratio of cache hit events to the total number of requests [162]. Fundamentally, a higher hit ratio is essential to obtain a better system efficiency. Several studies to achieve a significantly increased hit ratio have recently been proposed from the LVS perspective [26, 34, 117, 119, 139, 183, 187]. In particular, the work [117] investigated the field of view (FoV) aware algorithm for ABS-based edge caching served LVS services, where a common-FoV probabilistic model was analyzed based on the

viewing histories of previous users to improve the caching hit ratio. Their experiments demonstrated that their proposal significantly increases the hit ratio by at least 40% and 17% with respect to two conventional algorithms, the least frequently used (LFU) and the least recently used (LRU) [62], respectively. Subsequently, Maniotis *et al.* presented a smart edge caching algorithm for LVS in [119], from which their proposed performance in terms of the hit ratio is better than LFU, LRU, and first in first out (FIFO) algorithms. The size-popularity-layer-FoV (SPLF) strategy was proposed in [183], where (*i*) this strategy was dedicated to SVC-assisted LVS and (*ii*) the cache value of video chunks was estimated based on its size, popularity, SVC layer, and FoV existence. The results of [183] confirmed that its achievable hit ratio outperforms LFU, LRU, and greedy-dual size frequency (GDSF) [28] strategies. Meanwhile, a proposal in [26] formulated a caching problem of maximizing the cache hit ratio under the constraints of the storage capacity has revealed significant gains over the hit ratio comparison of LFU, LRU, and weighted GDSF [174]. In addition, in [187], the authors examined the max–min video utility fairness caching (MUFC) algorithm that achieves a better hit ratio than the advanced FIFO caching and FairRide caching [140]. Furthermore, in [139], Poularakis *et al.* studied the layer-aware cooperative caching (LCC) strategy with an effort to improve the hit ratio value for LVS services compared with independent caching and Femto caching [150]. In addition, the tradeoff between the hit ratio and content quality is also a considerable problem that has been addressed by the authors in [34] and resolved by their proposal, referred to as the hit ratio and content quality balancing algorithm, or HITCOT, where an edge caching system is considered for video-based multi-streaming ABS services.

5.7 Resource Consumption

Resource consumption has become a critical issue in any system, which has piqued the tremendous interest of many scientists around the world. This is because an LVS service is one of the most resource-hungry applications. In this study, we investigated resource consumption in four key aspects, including computing, caching, bandwidth, and energy, where the consumption of various resources in the LVS is interdependent. As a result, the tradeoffs among these resources' consumption are worth considering.

As mentioned previously, the work in [57] confirmed the improvement of the clients' bandwidth utilization and simultaneously minimizes the fluctuation of video quality based on the cooperative server–client HTTP in ABS-based LVS services. Meanwhile, [169] and [190] demonstrated that the bandwidth consumption of the LVS experience was reduced significantly by invoking edge/fog architectures. Zhang *et al.* considered less spectrum resource allocation, user scheduling, and transcoding decision problems for LVS services in heterogeneous networks [188], where the edge architecture was further leveraged. The system-level simulation results in [188] demonstrated that the effectiveness of low computing consumption, low latency, and high video quality is satisfied concurrently. In the same manner as [169, 188, 190], Rigazzi *et al.* exhibited the worthwhile deployment of edge/fog-based streaming [144], in which the degradation of 7% computing load, 27.3% caching memory usage, 3.6% energy consumption, up to 33% backhaul, and 5% fronthaul communication bandwidth was observed. In addition, viewport prediction was proposed in [47] to save the bandwidth resource, which is conducted via two key approaches: (*i*) the utilization of ML based on the past viewing behavior of a large number of users and (*ii*) near-term viewport prediction based on the current viewing behavior of users in the streaming session. The predictive concept was also presented in [11] with resource allocation prediction to deliver energy-efficient video streaming, leading to substantial energy savings. Subsequently, [58] and [96] proposed MCDNN and Chameleon novel algorithms based on deep neural network (DNN) utilization, respectively, to frequently adapt configurations for LVS applications. The results in [58] have shown the effective degradation of caching and energy consumption aspects as well as satisfying the low-latency stringent requirement under the constraints of the

given computing accuracy for LVS services, whereas the experimental outcomes of [96] have confirmed a 30–50% improvement in computing resources and 20–50% higher computing accuracy.

Although the multicast (mCast) Internet protocol is beneficial for reducing resource consumption because a stream in LVS mCast is delivered to a group of clients simultaneously in a single transmission attempt, it has a static and rigid nature when mCast is used separately [65]. Motivated by resolving this drawback, in [101], the software-defined networking (SDN) architecture is cooperative with the LVS mCast approach, where mCast has proven its capability of more than 50% link utilization improvement and 0% network losses, leading to a degradation in bandwidth consumption. Further consideration [175] was made to show the additional integration among the network function virtualization (NFV), SDN, and mCast in various beneficial network applications, including online conferencing, LVS, event monitoring, etc., from which the network throughput was maximized while minimizing the computing and bandwidth resource consumption. Moreover, SDN and mCast were cooperative with the scalable ABS to further support the LVS applications and obtain intelligent and dynamic service provisioning, where the equivalent bandwidth effectiveness was confirmed [176]. Meanwhile, [40] investigated a video transcoding method for adaptive bitrate LVS, where LVS services are responsible for transcoding a large number of videos into various bitrate levels to adaptively stream to users. In the proposed work, the edge-assisted architecture incorporating the LVS ecosystem and mCast distribution were presented, which showed the extension to not only provide the bandwidth and energy resource efficiency but also ensure fairness and live capability. To further save network resources for LVS, instead of unicast or mCast separation, a hybrid architecture was reported in [7]. With the hybrid architecture deployment, the network not only outperformed the hit ratio, spectral efficiency, video quality, frame loss rate, initial buffering time, and number of re-buffering events, but also balances both unicast and mCast tradeoffs such as (*i*) the higher network load but lower energy consumption using unicast and (*ii*) the lower network load but higher energy consumption using mCast.

Many contemporary studies have recently focused on analyzing the optimization problems of resource allocation for LVS applications. By invoking the conventional cloud architecture for LVS, Li *et al.* [106] proposed a solution for the joint optimization of communication and computational resource allocation with the aim of maximizing the QoE objective function. Subsequently, a cloud-based P2P architecture was considered in [66], where the authors analyzed the optimal bandwidth allocation problem to provide a high degree of user satisfaction. A further consideration of the edge cloud-based paradigm and VFN support for LVS experience was conducted in [23], in which the QoE objection was maximized under the load-balancing constraints of limited cloud computing and caching resources, transcoding requirements, throughput, and latency. As indicated in [110], the capability of the MEC and flexible transcoding ABS coordination has demonstrated its low-latency outperformance under limited computing, caching, and bandwidth resources. Simultaneously, in this contribution, the optimization problems were further considered in (*i*) joint optimization of access control and resource allocation and (*ii*) joint optimization of caching decision and transcoding strategies. In [107], the total expected energy consumption in an LVS service was minimized via the MEC support along with caching, transcoding, backhaul retrieving, and ABS platforms. The results obtained from [107] show not only the optimal energy scheme but also the effectiveness of the cache hit ratio. In [178], an online learning algorithm without training phases was proposed to actively estimate user preferences according to user feedback based on regression analysis, from which the optimal edge resource allocation strategy regarding computing, caching, and bandwidth parameters for MEC-based LVS services was developed. Unlike [23, 66, 106, 107, 110, 178], without cloud/edge platforms, Erfanian *et al.* [43] have recently introduced an optimizing available resource utilization strategy that focuses on the bandwidth resource for LVS based on SDN, NFV, and mCast support, where the requirement of the E2E latency threshold is satisfied.

5.8 Security and Privacy

Undoubtedly, the provision of user authentication and encryption should be enabled to secure the E2E secretary, avoid hacking and wiretapping for real-time streaming data, which is one of the foremost concerns in any system. Recently, blockchain technology has been proven capable of guaranteeing the security of E2E communications [70, 131]. With a focus on LVS, the blockchain differs from existing LVS-supporting technologies (e.g., cloud/edge-based, CDN, SDN, and VFN), where each stream information created by the communication between any two devices, referred to as transaction information, is stored in a chain block [6, 15, 100, 112, 129]. All transactions are visible at any node in the committed chain, which means that all modifications are tracked publicly. In this way, it helps the system to prevent cybercrimes, which guarantees the system's security. Furthermore, the blockchain uses asymmetric cryptography that includes public and private keys, where these keys are randomly created by strings of numbers [6, 15, 100, 112, 129]. Within such a large number of keys, it is mathematically impossible to deceptively gain access or guess the keys of other users, and security and privacy become stronger. For example, Li *et al.* [112] proposed MEC-assisted transcoding for blockchain-based live/on-demand video streaming while adapting the block size of blockchains, which significantly affects the performance. In addition, the alternating direction method of multipliers and smart contracts are enabled to facilitate the joint optimization of video transcoding offloading scheduling, block size adaptation, and resource allocation. In [129], the authors leveraged the help of an interplanetary file system (IPFS), HLS, and blockchain-based smart contracts to provide authentication, authorization, accessibility, and security for the LVS system. Meanwhile, Allen and Lucchi [6] considered the blockchain-based Red5-Network, which utilizes the Red5Coin token to make the network node transactions and further supports encrypted LVS streams to ensure content access allowed parties. Khalaf *et al.* [100] presented a new algorithm for blockchain-based LVS that comprised block architecture and cryptographic operations, from which it was confirmed its flexibility and scalability to effortlessly adapt to other platforms, such as Internet of Things (IoT), artificial intelligence, ML, and cloud/edge-based technologies. From the current market perspective, the seven biggest blockchain providers, including dlive, livepeer, Theta, VideoCoin, flixxo, LBRY, and Play2Live, were also surveyed in [15], where these companies have furnished not only on-demand video streaming but also the LVS platform. Despite the security and privacy contributions from highly efficient blockchain technologies, these approaches suffer from several fundamental limitations, including a consensus mechanism that consumes significant energy, considerable latency from transaction confirmation, and restricted scalability [67].

On the other hand, Varghese *et al.* [141] exhibited a data privacy platform based on hierarchical inner product encryption, shortened by HIPE, and broker with an anonymous pubsub architecture for LVS systems. The results in [141] have shown the security and privacy outperformance of their proposal compared with a system without HIPE. In [38], the practical privacy-preserving live streaming, called P3LS, was first proposed to protect the privacy of multiple streams in P2P LVS, where the evaluation of P3LS not only showed the privacy contribution but also 30% less bandwidth consumption than the non-P3LS strategy. Because the energy issue has become crucial in the mobile platform, Samet *et al.* [146] investigated the energy consumption comparison among the triple data encryption standard (3DES), advanced encryption standard (AES), and Blowfish algorithms for video streaming services. Unlike blockchain using asymmetric cryptography, DES, AES, and Blowfish are symmetric-key block ciphers that also enable security and privacy for the considered systems based on the various long key lengths. Furthermore, the privacy-aware architecture utilizes the face recognition framework to further enhance the secure characteristics of LVS [168], which has demonstrated safety and high accuracy.

Table 4. Summary of LVS references on performance metrics (Part 1).

Aspect	Ref.	Highlights	Main contributions
SA	[17, 32]	Non-redundant nodes for clouds	Obtained 104.24 h and 100.11 h of the annual downtime from [32] and [17], respectively
	[122, 123]	Redundant nodes for clouds	Achieved annual downtime of 83.798 h and 49.05 h from [123] and [122], respectively
	[9]	Software aging occurrence, cloud computing	Achieved only 33.69 minutes and 33.59 minutes of annual downtime from the time-based and prediction-based rejuvenation strategies, respectively
Video Bitrate	[57]	Cooperative client-server ABS	Provided high-quality LVS by achieving effective bandwidth usage and bitrate switches
	[149]	CDN, ABS using HLS	Achieved 11.58% improvement in throughput and 0.25% degradation in packet loss ratio
	[5]	CDN, ABS using Hampel filter	Achieved better 20% true positive rate than baselines
	[102]	Large-scale CDN	Achieved a better than 70% throughput than conventional benchmarks
E2E Latency	[108]	HTTP/2-based LVS	Improved video quality, smoothness, and E2E latency performance
	[105]		Packet buffering delay was minimized to 2 seconds
	[151]	A novel ABS proposal	Obtained closed-form expression for the average buffering delay
	[171]	MultiLevel ABS algorithm	Achievable E2E latency was reduced to only 0.1 second
	[156]	ABS using DRL	Estimated optimal video bitrate for ultra-low-latency intention
	[69]	ABS using QARC and DRL	Improved video quality by 18–25 % and decreased the E2E latency by 23–45 %
	[98, 134]	ABS using SVC	Provided coding bitrate decrements by 38% leading to a reduction in the E2E latency
	[103]	ABS using HEVC	Achieved approximately double encoding efficiency of the SVC with 56.7%
	[145]	ABS using SHVC	Gained around 20% decoding speedup in significantly diminishing the E2E latency
	[142]	ABS using VVC	Provided a low buffering delay of approximately 0.21 seconds
	[18]	Edge computing	Minimize the E2E latency and response time as well as provide the resource savings
	[177]	End–edge–cloud coordination	Satisfied low-latency, accurate analytic, LVS quality, and computing resource constraints
	[10]	Fog architecture	Low-latency support as well as ultra-reliability communication
	[110]	MEC, flexible transcoding ABS	Latency of 15–75 ms under limited computing, caching, and bandwidth constraints
Network QoS/QoE	[48]	Adaptive SDN architecture	Achieved end-user QoE improvement and high-accuracy MOS estimation
	[5]	ABS with large-scale CDN	Offered 92% accuracy in QoE (e.g., RoB, BPoVS, RoF, and APB) detection and 20% improvement in the true positive rate as compared with baselines
	[143]	ABS with GVBR	Achieved both comparable bitrate and cutting video interruption incidents by 90%
	[51]	ETHLE with 4K-resolution	Guaranteed QoE (e.g., initial startup delay, RoB, and latency), addressed the conventional transport-layer bottleneck with high data rate requirements
	[180]	QoE-driven resource allocation	Achieved QoE improvement, average mean time to failure, average PSNR, average energy consumption
	[111]	QoE-driven HASCP	Obtained QoE maximization and high bandwidth utilization
	[25]	Joint optimization problem	Achieved remarkable enhancement of the average QoE per use
	[37]	Hybrid NDN-IP	Provided the efficient utilization of network resources and improved QoE (e.g., APB, BRS, RoB, and spectrum)
	[159]	Deeplive	Achieved both the low execution training time and 15 − 55% improvement of QoE (e.g., RoB, FVQ, BRS, frame skipping, and latency)
	[189]	Online actor-critic learning	Maximized the QoE function under the time-delay constraint
	[116]	ReCLive	Distinguished live streams from video-on-demand streams by media-request patterns and effectively inferred QoE measurements (e.g., resolution and RoB)
	[30]	Innovative ML-based scheduling	Confirmed the effectiveness of QoS provisioning, packet loss rate, PSNR, and throughput
System Service-ability	[97]	ABS using FESTIVE	Provided 50% service stability, 40% fairness, and 10% efficiency improvements compared with various competitive commercial players
	[109]	ABS using PANDA	Increased service stability of 75%, improved fairness and efficiency performance
	[93]	A novelty ABS proposal	Adapted the received bandwidth to logarithmically converge to the estimated bandwidth, leading to all stability, efficiency, and fairness improvements
	[133]	ABS using SHANZ	Better balance between stability and efficiency than FESTIVE, PANDA, and AAASH
	[42]	ABS using ESTC	Improved performance of both the FESTIVE and PANDA strategies with respect to stability, efficiency, and fairness metrics
	[191]	ABS using TFDASH	

5.9 Summary and Discussion

In brief, several key performance pillars for LVS services, including SA, video bitrate, E2E latency, system serviceability, hit ratio, resource consumption, and security and privacy have been reviewed. Table 4 summarizes the key points of performance metrics for LVS services as well as representative references and their contributions. In particular, Section 5.1 presents the ability of LVS to provide satisfactory services to users whenever and wherever required, from which several of the most relevant publications by considering with/without the redundant node architecture and software aging issues in the web browser plug-in for cloud-based LVS services are surveyed in an attempt to either upgrade the SA value or degrade the downtime per year. With a focus on vision quality of LVS experience, Section 5.2 considers

Table 5. Summary of LVS references on performance metrics (Part 2).

Aspect	Ref.	Highlights	Main contributions
Hit Ratio	[117]	Edge, ABS using FoV-aware	Increased hit ratio by at least 40% and 17% compared with LFU and LRU, respectively
	[119]	Smart edge caching	Performance in terms of hit ratio was better than LFU, LRU, and FIFO
	[183]	Edge, SVC, SPLF caching	Achievable hit ratio outperformed LFU, LRU, and GDSF
	[26]	A novelty caching solution	Maximize the cache hit ratio under the constraints of the storage capacity and manifested the outperformance compared with LFU, LRU, and weighted GDSF
	[187]	MUFC caching	Gained better hit ratio than advanced FIFO caching and FairRide caching
	[139]	LCC caching	Significantly improved hit ratio value compared with independent and femto-caching
	[34]	Edge, ABS using HITCOT	Indicates the tradeoff between the hit ratio and content quality
Resource Consumption	[169, 190]	Edge/Fog architectures	Significantly reduced bandwidth consumption in a crowded network
	[188]	Edge computing	Gained effectiveness of low computing consumption, low latency, and high video quality
	[144]	Edge/fog architectures	Obtained the degradation of 7% computing load, 27.3% caching memory usage, 3.6% energy consumption, up to 33% backhaul, and 5% fronthaul communication bandwidth
	[47]	ML-based viewport prediction	Further saves bandwidth resources
	[11]	ML-based resource allocation	Energy efficiency approach to achieve substantial energy savings
	[58]	DNN-based MCDNN	Effective degradation of caching and energy consumption satisfied the low-latency stringent requirement under the constraints of the given computing accuracy
	[96]	DNN-based Chameleon	30–50 % computing resource improvement and 20–50 % higher computing accuracy
	[101]	SDN and mCast	Improved 50% link utilization achieved 0% network losses, leading to bandwidth savings
	[175]	NFV, SDN, and mCast	Maximized network throughput and minimized computing and bandwidth consumption
	[176]	SDN, mCast, and scalable ABS	Confirmed equivalent bandwidth effectiveness
	[40]	Edge, mCast, and ABS	Gained bandwidth and energy resource efficiency, ensured fairness and live capability
	[7]	Hybrid unicast and mCast	Provided high network load, low energy consumption, outperformed in the hit ratio, spectral efficiency, video quality, frame loss rate, buffering delay, re-buffering number
	[106]	Conventional cloud	Joint optimization of communication and computational resource allocation
	[66]	Cloud-based P2P	Optimal bandwidth allocation problem to provide high satisfaction degrees for users
	[23]	Edge and VFN	Maximized QoE under the load-balancing constraints of limited cloud computing and caching resources, transcoding requirements, throughput, and latency
	[110]	MEC, flexible transcoding ABS	Joint optimization of access control and resource allocation, joint optimization of caching decisions and transcoding strategies, and optimal latency on limited resource constraints
	[107]	MEC, ABS, backhaul retrieve	Optimal energy consumption and effectiveness of the cache hit ratio
	[178]	MEC, no-train online learning	Optimal edge resource allocation under limited computing, caching, and bandwidth
	[43]	SDN, NFV, and mCast	Optimal available-bandwidth resource utilization satisfies the E2E latency threshold
Security and Privacy	[112]	Blockchain, MEC	Joint optimization of video coding offloading, block size, and resource allocation
	[129]	Blockchain, HLS, IPFS	Provided authentication, authorization, accessibility, and security for the LVS system
	[6]	Blockchain, Red5-Network	Utilized the Red5Coin token for transactions and supported encrypted LVS streams
	[100]	Novel blockchain proposal	Effortlessly adapted to IoT, artificial intelligence, ML, and cloud/edge-based technologies
	[15]	Current blockchain providers	Surveyed live, livepeer, theta, videocoin, flixxo, LBRY, and Play2Live providers
	[141]	Secure concepts and frameworks	Proposed the HIPE algorithm and broker with an anonymous pubsub architecture to provide security and privacy outperformance
	[38]		Proposed P3LS algorithm to protect the privacy of multiple streams in P2P LVS
	[146]		Investigated the energy consumption comparison among 3DES, AES, and Blowfish
	[168]		Utilized the face recognition of privacy-aware architecture to further enhance security

video bitrate issues, where many ABS techniques have been adapted to achieve the highest video quality possible in the context of bandwidth fluctuations owing to changing network conditions. Nonetheless, the ABS techniques in Section 5.2 do not thoroughly consider the E2E latency aspect, whereas the nature of LVS systems comes from the stringent constraints on real-time providability. Section 5.3 considers further cooperative research between ABS mechanisms and HTTP/2, DRL, video coding standards, and hierarchical computing models, which proved its capability in terms of E2E latency and video quality for LVS, where SVC, HEVC, SHVC, VVC, and hybrid architectures can be listed as helpful mechanisms for video coding standards. In Section 5.4, we provide the network QoS/QoE aspect that reflects the relationship between the technology provisioning and the end-users satisfactory, where QoS/QoE in terms of MOS, RoB, RPoVS, RoF, APB, FVQ, BPS, frame skipping, resolution, latency, spectrum, etc. are beneficial in measurements and estimation. Since ML-based applications have recently become very popular because of their powerful characteristics, network QoS/QoE measurements with the ML-based prediction are also investigated in Section 5.4. The serviceability of the LVS system with respect to the service stability, efficiency, and fairness metrics is provided in Section 5.5, where

the FESTIVE, PANDA, SHANZ, ESTC, and TFDASH strategies are beneficial. In Section 5.6, various novel algorithms, including FoV-aware, smart edge caching, SPLF, MUFC, LCC, and HITCOT have been invoked to significantly improve the hit ratio value for LVS systems compared with several conventional benchmarks. Because an LVS service is one of the most resource-hungry applications, the survey scope of Section 5.7 is covered within four interdependent measurements: computing, caching, bandwidth, and energy. In addition to the efficiency achieved by mCast, NFV, SDN, DNN, and ABS approaches, the optimization problems among one or several of the four key parameters were analyzed. In Section 5.8, we reviewed the security and privacy perspectives within the LVS scope, which are based on blockchain technologies as well as non-blockchain platforms. It is worth noting that the integration of the aforementioned algorithms has resolved only a few performance aspects; there were some tradeoffs regarding providability.

6 OPEN CHALLENGES

6.1 System Scalability

Massive connectivity has been considered one of the major requirements for realizing future communication networks, where billions of user devices participate in the Internet to exchange information [172]. As video traffic is increasingly dominant in 5G ecosystems and beyond, LVS frameworks should provide scalability to adaptively serve a massive number of user requests with various streaming flows simultaneously. Because user interests are spatiotemporal patterns, LVS capabilities must be flexibly elastic to any fluctuations of service request volumes and distributions in both the time and space domains. For instance, a self-organized model of LVS frameworks automatically activates/deactivates LVS-aware functions at several network components within an optimal design to achieve energy and computation efficiencies while retaining service quality. Conversely, SA and video bitrate can be considered in a tradeoff optimization to balance these catch-22 features. Obviously, scalability is critical for optimal and efficient LVS systems in the current and next communication network generations; therefore, this capability deserves the attention of research communities.

6.2 High Video Bitrate

Recently, advanced electronic technologies have enabled new generations of display resolutions with extremely high pixel density in a single screen panel, such as 4K (4096 horizontal pixels), 8K (7680 horizontal pixels), and 10K (10,240 horizontal pixels). Consequently, the bitrates of video streams must be proportionally increased for optimal exploitation of these display resolutions [61]. A broad range of video resolutions (from 144p to 10K) should be offered by LVS systems to satisfy various user devices. For example, roadside notification screens in smart transportation systems display traffic conditions within low-resolution video streams; digital advertisement posters broadcast high-resolution video clips. Online gaming, live video, and video-on-demand services typically offer the highest resolution with their best efforts. In these scenarios, LVS systems are required to optimize streaming algorithms by considering available communication, computation, and storage resources at every LVS-aware network component to efficiently provide various bitrates simultaneously. A high video bitrate significantly consumes resources; therefore, it is essential to provide energy-efficient streaming and transcoding solutions.

6.3 Latency-sensitive Applications

Numerous video-stream-based applications such as remote healthcare, online multiplayer games, and precise manufacturing controls are considered latency-sensitive services; they require ultra-reliable and low-latency communications. Conversely, to extend the mobile coverage in the context of 5G and beyond, aerial access infrastructure constituted by

airborne access points such as UAVs and satellites provide Internet connections to end users in underserved areas with considerable latency [35]. Hence, minimizing E2E latency and stalling events in such networking environments is critical for LVS systems to offer smooth playback. Among the applicable strategies, efficient caching policies and transcoding models help greatly in achieving the target. To this end, a deeper understanding of user behaviors with advanced AI techniques for video interest prediction is necessary and requires further study to improve LVS system performance. It is worth noting that latency minimization can be jointly resolved in several tradeoff problems considering other metrics such as video bitrate, SA, and security.

6.4 Intelligent Recommendations

Advanced AI technologies have been driving multiple features of LVS systems, whereas intelligent video recommendation is important for engaging users with content providers [27]. An effective recommendation is essential to offer users with relevant video content, which are expected to satisfy user interests. Depending on the scope, the characteristics of user behaviors should be investigated differently. For instance, a large-scale LVS system concerns geographical user request distribution and density, whereas a small-scale LVS system offers video recommendations based on the demographic locality of users' gender, age, occupation, movie genre, and time patterns. Obviously, an intelligent recommendation feature can only be developed if the system has appropriate knowledge of user behaviors and expectations. This problem is considered more challenging in this era, where digital content is produced every second and published on the Internet. Therefore, efficient learning and fusing of multiple aspects of user behaviors should be a focus of future research on LVS development.

6.5 Automatic Editing

As live videos do not accept delays in streaming video content to the Internet, a comprehensive production process is inapplicable to LVS systems. By contrast, automatic editing actions are preferred for live video streams. For instance, auto-generated caption insertion was introduced to enrich YouTube Live services, whereas Facebook messenger applications enable real-time filters and augmented objects into video calls. In [185], a novel autoremover tool was proposed to automatically remove unwanted objects from autonomous driving videos. These services are prime examples of automatic editing features enabled in LVS systems. Although such basic features have significantly improved LVS quality, more advanced and intelligent editing effects should be provided by adequately exploiting powerful AI techniques and in-network computation capabilities.

7 CONCLUDING REMARKS

In this paper, we have provided a contemporary survey on LVS from a computation-driven perspective, where in-network computation capabilities play a key role in assisting LVS operations. By conducting a thorough investigation of LVS from multiple aspects, we have constructed a reference framework for interested readers with state-of-the-art knowledge about LVS systems. In particular, LVS commercial platforms, standard architectures, service models, and performance metrics have been analyzed to obtain valuable insights and discussions. Based on these observations, we have highlighted open research challenges in LVS for future studies.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1G1A1008105). The work of Anh-Tien Tran and Sungrae Cho was supported by the National Research

Foundation of Korea (NRF) under Grant NRF-2019R1A2C1090447 funded by the Korea Government (Ministry of Science and ICT). The work of Tran Thien Thanh and Vo Nguyen Quoc Bao was funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant 102.02-2018.320. Vo Nguyen Quoc Bao and Sungrae Cho are corresponding authors.

REFERENCES

[1] Flowplayer AB. 2022. Flowplayer: The Performance First Online Video Platform. Retrieved Jan. 11, 2022 from https://flowplayer.com/

[2] Miran Taha Abdullah Abdullah, Jaime Lloret, Alejandro Cánovas Solbes, and Laura García-García. 2017. Survey of transportation of adaptive multimedia streaming service in Internet. *Network Protocols and Algorithms* 9, 1-2 (2017), 85–125.

[3] Adobe System Inc. 2021. *HTTP Dynamic Streaming*. https://www.adobe.com/products/hds-dynamic-streaming.html

[4] Samira Afzal, Vanessa Testoni, Christian Esteve Rothenberg, Prakash Kolan, and Imed Bouazizi. 2019. A holistic survey of wireless multipath video streaming. *arXiv preprint arXiv:1906.06184* (Jun. 2019).

[5] Adnan Ahmed, Zubair Shafiq, Harkeerat Bedi, and Amir Khakpour. 2017. Suffering from buffering? Detecting QoE impairments in live video streams. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, Toronto, ON, 1–10.

[6] Chris Allen and Davide Lucchi. 2019. Red5 network: decentralized real-time secure video streaming service. In *Proceedings of the 10th ACM Multimedia Systems Conference*. Amherst, Massachusetts, 296–299.

[7] Saleh Almowuena, Md Mahfuzur Rahman, Cheng-Hsin Hsu, Ahmad AbdAllah Hassan, and Mohamed Hefeeda. 2016. Energy-aware and bandwidth-efficient hybrid video streaming over mobile networks. *IEEE Transactions on Multimedia* 18, 1 (Jan. 2016), 102–115.

[8] Apple Inc. 2021. *HTTP Live Streaming*. https://developer.apple.com/streaming/

[9] Jean Araujo, Felipe Oliveira, Rubens de S Matos, Matheus Torquato, Joao Ferreira, and Paulo Romero Martins Maciel. 2016. Software Aging Issues in Streaming Video Player. *Journal of Software* 11, 6 (Jun. 2016), 554–568.

[10] Bouchaib Assila, Abdellatif Kobbane, Mohammed El Koutbi, Jalel Ben-Othman, and Lynda Mokdad. 2018. Caching as a Service in 5G Networks: Intelligent Transport and Video on Demand Scenarios. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Abu Dhabi, United Arab Emirates, 1–6.

[11] Ramy Atawia, Hossam S Hassanein, and Aboelmagd Noureldin. 2017. Energy-efficient predictive video streaming under demand uncertainties. In *2017 IEEE International Conference on Communications (ICC)*. IEEE, Paris, France, 1–6.

[12] Emna Baccour, Aiman Erbad, Kashif Bilal, Amr Mohamed, Mohsen Guizani, and Mounir Hamdi. 2020. FacebookVideoLive18: A live video streaming dataset for streams metadata and online viewers locations. In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. 476–483.

[13] Emna Baccour, Aiman Erbad, Amr Mohamed, Fatima Haouari, Mohsen Guizani, and Mounir Hamdi. 2020. RL-OPRA: Reinforcement Learning for Online and Proactive Resource Allocation of crowdsourced live videos. *Future Generation Computer Systems* 112 (2020), 982–995.

[14] Alcardo Alex Barakabitze, Nabajeet Barman, Arslan Ahmad, Saman Zadtootaghaj, Lingfen Sun, Maria G Martini, and Luigi Atzori. 2019. QoE management of multimedia streaming services in future networks: A tutorial and survey. *IEEE Communications Surveys & Tutorials* 22, 1 (2019), 526–565.

[15] Nabajeet Barman, GC Deepak, and Maria G Martini. 2020. Blockchain for Video Streaming: Opportunities, Challenges, and Open Issues. *Computer* 53, 7 (Jul. 2020), 45–56.

[16] Abdelhak Bentaleb, Bayan Taani, Ali C Begen, Christian Timmerer, and Roger Zimmermann. 2019. A survey on bitrate adaptation schemes for streaming media over HTTP. *IEEE Communications Surveys & Tutorials* 21, 1 (Firstquarter 2019), 562–585.

[17] Maria Clara Bezerra, Rosangela Melo, Jamilson Dantas, Paulo Maciel, and Francisco Vieira. 2014. Availability modeling and analysis of a VoD service for eucalyptus platform. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. San Diego, CA, USA, 3779–3784.

[18] Kashif Bilal and Aiman Erbad. 2017. Edge computing for interactive media and video streaming. In *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, Valencia, Spain, 68–73.

[19] K. Bilal, A. Erbad, and M. Hefeeda. 2017. Crowdsourced multi-view Live Video Streaming using Cloud Computing. *IEEE Access* 5 (2017), 12635–12647.

[20] Kashif Bilal, Aiman Erbad, and Mohamed Hefeeda. 2018. QoE-aware distributed cloud-based live streaming of multisourced multiview videos. *Journal of Network and Computer Applications* 120 (2018), 130–144.

[21] Brightcove. [n.d.]. Brightcove Inc. Retrieved Jan. 11, 2022 from https://www.brightcove.com/en/

[22] Inc. Brightcove. 2022. Video JS. Retrieved Jan. 11, 2022 from https://videojs.com/

[23] Utku Bulkan, Muddesar Iqbal, and Tasos Dagiuklas. 2018. Load-balancing for edge QoE-based VNF placement for OTT video streaming. In *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, Abu Dhabi, United Arab Emirates, 1–6.

[24] Alexander Bychok. 2020. What is video bitrate: Full guide. Retrieved Jan. 11, 2022 from https://restream.io/blog/what-is-video-bitrate

[25] Wei-Yu Chen, Po-Yu Chou, Chih-Yu Wang, Ren-Hung Hwang, and Wen-Tsuen Chen. Jun. 2021. Dual Pricing Optimization for Live Video Streaming in Mobile Edge Computing with Joint User Association and Resource Management. *IEEE Transactions on Mobile Computing* (Jun. 2021).

[26] Xing Chen, Lijun He, Shang Xu, Shibo Hu, Qingzhou Li, and Guizhong Liu. 2019. Hit ratio driven mobile edge caching scheme for video on demand services. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, Shanghai, China, 1702–1707.

[27] Xusong Chen, Dong Liu, Zhiwei Xiong, and Zheng-Jun Zha. 2020. Learning and fusing multiple user interest representations for micro-video and movie recommendations. *IEEE Transactions on Multimedia* 23 (2020), 484–496.

[28] Ludmila Cherkasova. 1998. *Improving WWW proxies performance with greedy-dual-size-frequency caching policy.* Hewlett-Packard Laboratories.

[29] Clappr. [n.d.]. Clappr: An extensible media player for applications. Retrieved Jan. 11, 2022 from http://clappr.io/

[30] Ioan-Sorin Comşa, Gabriel-Miro Muntean, and Ramona Trestian. Mar. 2021. An innovative machine-learning-based scheduling solution for improving live UHD video streaming quality in highly dynamic network environments. *IEEE Transactions on Broadcasting* 67, 1 (Mar. 2021), 212–224.

[31] Dacast. [n.d.]. Live Streaming & Video Hosting Platform. Retrieved Jan. 11, 2022 from https://www.dacast.com

[32] Jamilson Dantas, Rubens Matos, Jean Araujo, Danilo Oliveira, Andre Oliveira, and Paulo Maciel. 2016. Hierarchical model and sensitivity analysis for a cloud-based VoD streaming service. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*. IEEE, Toulouse, France, 10–16.

[33] Nhu-Ngoc Dao, Woongsoo Na, and Sungrae Cho. 2020. Mobile cloudization storytelling: Current issues from an optimization perspective. *IEEE Internet Computing* 24, 1 (2020), 39–47.

[34] Nhu-Ngoc Dao, Duy Trong Ngo, Ngoc-Thanh Dinh, Trung V Phan, Nam D Vo, Sungrae Cho, and Torsten Braun. 2021. Hit Ratio and Content Quality Tradeoff for Adaptive Bitrate Streaming in Edge Caching Systems. *IEEE Systems Journal* 15, 4 (2021), 5094–5097.

[35] Nhu-Ngoc Dao, Quoc-Viet Pham, Dinh-Thuan Do, and Schahram Dustdar. 2021. The sky is the edge–Toward mobile coverage from the sky. *IEEE Internet Computing* 25, 2 (2021), 101–108.

[36] Nhu-Ngoc Dao, Quoc-Viet Pham, Ngo Hoang Tu, Tran Thien Thanh, Vo Nguyen Quoc Bao, Demeke Shumeye Lakew, and Sungrae Cho. 2021. Survey on aerial radio access networks: Toward a comprehensive 6G access infrastructure. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 1193–1225.

[37] Ishita Dasgupta, Susmit Shannigrahi, and Michael Zink. 2021. A hybrid NDN-IP Architecture for Live Video Streaming: A QoE Analysis. In *2021 IEEE International Symposium on Multimedia (ISM)*. Naple, Italy, 148–157.

[38] Jérémie Decouchant, Antoine Boutet, Jiangshan Yu, and Paulo Esteves-Verissimo. 2019. P3LS: Plausible deniability for practical privacy-preserving live streaming. In *2019 38th Symposium on Reliable Distributed Systems (SRDS)*. IEEE, Lyon, France, 1–109.

[39] Google Developers. 2022. WebRTC: Real-time communication for the web. Retrieved Jan. 11, 2022 from https://webrtc.org/?hl=en

[40] Pradeep Dogga, Sandip Chakraborty, Subrata Mitra, and Ravi Netravali. 2019. Edge-based transcoding for adaptive live video streaming. In *2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19)*. Renton, WA.

[41] C. Dong, W. Wen, T. Xu, and X. Yang. 2019. Joint Optimization of Data-Center Selection and Video-Streaming Distribution for Crowdsourced Live Streaming in a Geo-Distributed Cloud Platform. *IEEE Transactions on Network and Service Management* 16, 2 (2019), 729–742.

[42] Oussama El Marai, Tarik Taleb, Mohamed Menacer, and Mouloud Koudil. 2018. On improving video streaming efficiency, fairness, stability, and convergence time through client–server cooperation. *IEEE Transactions on Broadcasting* 64, 1 (Mar. 2018), 11–25.

[43] Alireza Erfanian, Farzad Tashtarian, Anatoliy Zabrovskiy, Christian Timmerer, and Hermann Hellwagner. 2021. OSCAR: On Optimizing Resource Utilization in Live Video Streaming. *IEEE Transactions on Network and Service Management* (Mar. 2021).

[44] European Telecommunications Standard Institute (ETSI) . 2013. Universal Mobile Telecommunications System (UMTS); LTE; Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (3GPP TS 26.247 version 11.1.0 Release 11) . Sophia-Antipolis Cedex, France.

[45] European Telecommunications Standard Institute (ETSI). 2009. Universal Mobile Telecommunication System (UMTS); LTE; Transparent end-to-end Packet-Switched Streaming Service (PSS); Protocols and Codecs. Sophia-Antipolis Cedex, France.

[46] Przemysław Falkowski-Gilski and Tadeus Uhl. 2020. Current trends in consumption of multimedia content using online streaming platforms: A user-centric survey. *Computer Science Review* 37 (2020), 100268.

[47] Xianglong Feng, Viswanathan Swaminathan, and Sheng Wei. 2019. Viewport prediction for live 360-degree mobile video streaming using user-content hybrid motion tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (Jun. 2019), 1–22.

[48] Miguel García-Pineda, Santiago Felici-Castell, and Jaume Segura-García. 2017. Adaptive SDN-based architecture using QoE metrics in live video streaming on Cloud Mobile Media. In *2017 Fourth International Conference on Software Defined Systems (SDS)*. IEEE, Valencia, Spain, 100–105.

[49] GB/T 17975.1-2010. 2010. Information Technology–Generic Coding Of Moving Pictures And Associated Audio Information–Part 1:Systems.

[50] GB/T 20090.1-2012. 2012. Information technology - Advanced coding of audio and video - Part 1: System.

[51] Chang Ge, Ning Wang, Wei Koong Chai, and Hermann Hellwagner. Aug. 2018. QoE-assured 4K HTTP live streaming via transient segment holding at mobile edge. *IEEE Journal on Selected Areas in Communications* 36, 8 (Aug. 2018), 1816–1830.

[52] Romeo Giuliano, Franco Mazzenga, and Alessandro Vizzarri. 2020. Integration of Broadcaster and Telco Access Networks for Real Time/Live Events. *IEEE Transactions on Broadcasting* 66, 3 (2020), 667–675.

[53] Google. 2022. Google Hangouts. Retrieved Jan. 11, 2022 from https://hangouts.google.com/

[54] W3C Working Group. 2016. ISO Common Encryption ('cenc') Protection Scheme for ISO Base Media File Format Stream Format. Retrieved Jan. 11, 2022 from https://www.w3.org/TR/eme-stream-mp4/

[55] Y. Guo, F. R. Yu, J. An, K. Yang, C. Yu, and V. C. M. Leung. 2020. Adaptive Bitrate Streaming in Wireless Networks With Transcoding at Network Edge Using Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology* 69, 4 (2020), 3879–3892.

[56] Haivision. 2022. Secure Reliable Transport (SRT) Protocol Technical Overview. Retrieved Jan. 11, 2022 from https://www.haivision.com/resources/white-paper/srt-protocol-technical-overview/?utm_campaign=SEO+Resources+SRT+Tech+Specs+PR

[57] Sangwook Han, Yunmin Go, Hyunmin Noh, and Hwangjun Song. 2019. Cooperative server-client HTTP adaptive streaming system for live video streaming. In *2019 International Conference on Information Networking (ICOIN)*. IEEE, 176–180.

[58] Seungyeop Han, Haichen Shen, Matthai Philipose, Sharad Agarwal, Alec Wolman, and Arvind Krishnamurthy. 2016. MCDNN: An approximation-based execution framework for deep stream processing under resource constraints. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. Singapore, Singapore, 123–136.

[59] F. Haouari, E. Baccour, A. Erbad, A. Mohamed, and M. Guizani. 2019. QoE-Aware Resource Allocation for Crowdsourced Live Streaming: A Machine Learning Approach. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. Shanghai, China, 1–6.

[60] F. Haouari, E. Baccour, A. Erbad, A. Mohamed, and M. Guizani. 2019. Transcoding Resources Forecasting and Reservation for Crowdsourced Live Streaming. In *2019 IEEE Global Communications Conference (GLOBECOM)*. Waikoloa, HI, USA, 1–7.

[61] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. 2020. Space-time-aware multi-resolution video enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2859–2868.

[62] Gerhard Hasslinger, Juho Heikkinen, Konstantinos Ntougias, Frank Hasslinger, and Oliver Hohlfeld. 2018. Optimum caching versus LRU and LFU: Comparison and combined limited look-ahead strategies. In *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, Shanghai, China, 1–6.

[63] Marc Helmold. 2021. New Work in Education and Teaching. In *New Work, Transformational and Virtual Leadership*. Springer, 143–155.

[64] HootSuite. 2021. *25 YouTube Statistics that May Surprise You: 2021 Edition.* https://blog.hootsuite.com/youtube-stats-marketers

[65] Mojtaba Hosseini, Dewan Tanvir Ahmed, Shervin Shirmohammadi, and Nicolas D Georganas. 2007. A survey of application-layer multicast protocols. *IEEE Communications Surveys & Tutorials* 9, 3 (Third quarter 2007), 58–74.

[66] Guowei Huang, Lingjing Kong, Keke Wu, and Zhi Chen. 2017. A Bandwidth allocation policy for helpers in cloud-assisted P2P video-on-demand systems. In *2017 Fifth International Conference on Advanced Cloud and Big Data (CBD)*. IEEE, Shanghai, China, 7–12.

[67] Junqin Huang, Linghe Kong, Guihai Chen, Min-You Wu, Xue Liu, and Peng Zeng. 2019. Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism. *IEEE Transactions on Industrial Informatics* 15, 6 (Jun. 2019), 3680–3689.

[68] S. Huang, X. Huang, and N. Ansari. 2021. Budget-aware Video Crowdsourcing at the Cloud-enhanced Mobile Edge. *IEEE Transactions on Network and Service Management* 18, 2 (2021), 2123–2137.

[69] Tianchi Huang, Rui-Xiao Zhang, Chao Zhou, and Lifeng Sun. 2018. QARC: Video quality aware rate control for real-time video streaming based on deep reinforcement learning. In *Proceedings of the 26th ACM international conference on Multimedia*. Association for Computing Machinery, Seoul, Republic of Korea, 1208–1216.

[70] Tam T Huynh, Thuc D Nguyen, and Hanh Tan. 2019. A survey on security and privacy issues of blockchain technology. In *2019 International Conference on System Science and Engineering (ICSSE)*. IEEE, Dong Hoi, Vietnam, 362–367.

[71] IEEE 1857.7-2018. 2018. IEEE Standard for Adaptive Streaming.

[72] IETF. 2003. The Base16, Base32, and Base64 Data Encodings. Retrieved Jan. 11, 2022 from https://tools.ietf.org/html/rfc3548

[73] IETF. 2003. UTF-8, a transformation format of ISO 10646. Retrieved Jan. 11, 2022 from https://tools.ietf.org/html/rfc3629

[74] IETF. 2006. RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals. Retrieved Jan. 11, 2022 from https://tools.ietf.org/pdf/rfc4733.pdf

[75] IETF. 2006. SDP: Session Description Protocol. Retrieved Jan. 11, 2022 from https://tools.ietf.org/html/rfc4566#page-10

[76] IETF. 2010. Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP). Retrieved Jan. 11, 2022 from https://tools.ietf.org/html/rfc5764

[77] IETF. 2011. RTP Payload Format for MPEG-4 Audio/Visual Streams. Retrieved Jan. 11, 2022 from https://tools.ietf.org/html/rfc6416

[78] IETF. 2012. Definition of the Opus Audio Codec. Retrieved Jan. 11, 2022 from https://tools.ietf.org/html/rfc6716

[79] IETF. 2016. Real-Time Streaming Protocol Version 2.0. Retrieved Jan. 11, 2022 from https://tools.ietf.org/html/rfc7826.html

[80] IETF. 2017. HTTP Live Streaming. Retrieved Jan. 11, 2022 from https://tools.ietf.org/html/rfc8216

[81] IETF. 2021. WebRTC Security Architecture. Retrieved Jan. 11, 2022 from https://tools.ietf.org/html/rfc8827

[82] Adobe Systems Inc. 2012. Adobe's Real Time Messaging Protocol. Retrieved Jan. 11, 2022 from https://www.adobe.com/content/dam/acom/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf

[83] Adobe Systems Inc. 2013. Action Message Format - AMF 3. Retrieved Jan. 11, 2022 from https://wwwimages2.adobe.com/content/dam/acom/en/devnet/pdf/amf-file-format-spec.pdf

[84] Bitmovin Inc. 2022. Bitmovin: Play everywhere. Retrieved Jan. 11, 2022 from https://bitmovin.com/video-player/

[85] Adobe Systems Incorporated. 2013. HTTP Dynamic Streaming Specification - Version 3.0 (FINAL). Retrieved Jan. 11, 2022 from https://wwwimages2.adobe.com/content/dam/acom/en/devnet/hds/pdfs/adobe-hds-specification.pdf

[86] Instagram. 2022. Instagram from Meta. Retrieved Jan. 11, 2022 from https://www.instagram.com/

[87] ISO/IEC 13818-1:2019. 2019. Information technology — Generic coding of moving pictures and associated audio information — Part 1: Systems.

[88] ISO/IEC 23009-1:2012. 2012. Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats.

[89] ISO/IEC 23009-1:2019. 2019. Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats.

[90] ISO/IEC Standard 23009-5. 2017. Information Technology—Dynamic Adaptive Streaming Over HTTP (DASH)—Part 5: Server and Network Assisted DASH (SAND).
[91] ITU-T Recommendation E.860 (06/2002). [n.d.]. *Framework of a Service Level Agreement.*
[92] ITU-T Recommendation X.140 (09/92). [n.d.]. *General Quality of Service Parameters for Communication via Public Data Networks.*
[93] Saba Qasim Jabbar, Dheyaa Jasim Kadhim, and Yu Li. 2018. Proposed an adaptive bitrate algorithm based on measuring bandwidth and video buffer occupancy for providing smoothly video streaming. *Technology* 9, 2 (2018), 191–195.
[94] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. 1984. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA* (1984).
[95] Behrouz Jedari, Gopika Premsankar, Gazi Illahi, Mario Di Francesco, Abbas Mehrabi, and Antti Ylä-Jääski. 2020. Video Caching, Analytics and Delivery at the Wireless Edge: A Survey and Future Directions. *IEEE Communications Surveys & Tutorials* 23, 1 (2020), 431–471.
[96] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica. 2018. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. Budapest, Hungary, 253–266.
[97] Junchen Jiang, Vyas Sekar, and Hui Zhang. 2012. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. 97–108.
[98] Yuxuan Jiang, Bo Sun, and Danny HK Tsang. 2021. Not Taken for Granted: Configuring Scalable Live Video Streaming Under Throughput Fluctuations in Mobile Edge Networks. *IEEE Transactions on Vehicular Technology* 70, 3 (2021), 2771–2782.
[99] Kaltura. 2021. Our mission is to power any video experience, for any organization. Retrieved Jan. 11, 2022 from https://corp.kaltura.com/
[100] Osamah Ibrahim Khalaf, Ghaida Muttashar Abdulsahib, Hamed Daei Kasmaei, and Kingsley A Ogudo. 2020. A new algorithm on application of blockchain technology in live stream video transmissions and telecommunications. *International Journal of e-Collaboration (IJeC)* 16, 1 (2020), 16–32.
[101] Ahmed Khalid, Ahmed H Zahran, and Cormac J Sreenan. 2017. mCast: An SDN-Based Resource-Efficient Live Video Streaming Architecture with ISP-CDN Collaboration. In *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. IEEE, Singapore, 95–103.
[102] Ahmed Khalid, Ahmed H Zahran, and Cormac J Sreenan. 2019. An SDN-based device-aware live video service for inter-domain adaptive bitrate streaming. In *Proceedings of the 10th ACM Multimedia Systems Conference*. Association for Computing Machinery, Amherst, Massachusetts, 121–132.
[103] Daisuke Kobayashi, Ken Nakamura, Tatsuya Osawa, Yuya Omori, Takayuki Onishi, and Hiroe Iwasaki. 2019. A Real-Time 4K HEVC Multi-Channel Encoding System with Content-Aware Bitrate Control. In *2019 IEEE Global Communications Conference (GLOBECOM)*. Waikoloa, HI, USA, 1–6.
[104] Jonathan Kua, Grenville Armitage, and Philip Branch. 2017. A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP. *IEEE Communications Surveys & Tutorials* 19, 3 (Thirdquater 2017), 1842–1866.
[105] Hung T Le, Thoa Nguyen, Nam Pham Ngoc, Anh T Pham, and Truong Cong Thang. 2018. HTTP/2 push-based low-delay live streaming over mobile networks with stream termination. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 9 (Sep. 2018), 2423–2427.
[106] Jie Li, Cong Zhang, Zhi Liu, Wei Sun, and Qiyue Li. 2020. Joint communication and computational resource allocation for QoE-driven point cloud video streaming. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, Dublin, Ireland, 1–6.
[107] Liang Li, Dian Shi, Ronghui Hou, Rui Chen, Bin Lin, and Miao Pan. 2020. Energy-efficient proactive caching for adaptive video streaming via data-driven optimization. *IEEE Internet of Things Journal* 7, 6 (Jun. 2020), 5549–5561.
[108] Yunlong Li, Shanshe Wang, Xinfeng Zhang, Chao Zhou, and Siwei Ma. 2020. High Efficiency Live Video Streaming With Frame Dropping. In *2020 IEEE International Conference on Image Processing (ICIP)*. Abu Dhabi, United Arab Emirates, 1226–1230.
[109] Zhi Li, Xiaoqing Zhu, Joshua Gahm, Rong Pan, Hao Hu, Ali C Begen, and David Oran. 2014. Probe and adapt: Rate adaptation for HTTP video streaming at scale. *IEEE Journal on Selected Areas in Communications* 32, 4 (Apr. 2014), 719–733.
[110] Chunyu Liu, Heli Zhang, Hong Ji, and Xi Li. 2021. MEC-assisted flexible transcoding strategy for adaptive bitrate video streaming in small cell networks. *China Communications* 18, 2 (Feb. 2021), 200–214.
[111] Junquan Liu, Weizhan Zhang, Shouqin Huang, Haipeng Du, and Qinghua Zheng. 2021. QoE-driven HAS Live Video Channel Placement in the Media Cloud. *IEEE Transactions on Multimedia* (2021).
[112] Mengting Liu, Yinglei Teng, F Richard Yu, Victor CM Leung, and Mei Song. 2020. A mobile edge computing (MEC)-enabled transcoding framework for blockchain-based video streaming. *IEEE Wireless Communications* 27, 2 (Apr. 2020), 81–87.
[113] Facebook Live. 2022. Meta for Media. Retrieved Jan. 11, 2022 from https://www.facebook.com/formedia/tools/facebook-live
[114] Youtube Live. 2022. YouTube Live Streaming & Premieres. Retrieved Jan. 11, 2022 from https://www.youtube.com/intl/en_us/howyoutubeworks/product-features/live/#youtube-live
[115] Vimeo Livestream. 2022. The world's only all-in-one video solution. Retrieved Jan. 11, 2022 from https://vimeo.com/vimeolivestream
[116] Sharat Chandra Madanapalli, Alex Mathai, Hassan Habibi Gharakheili, and Vijay Sivaraman. 2021. ReCLive: Real-Time Classification and QoE Inference of Live Video Streaming Services. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*. IEEE, Tokyo, Japan, 1–7.
[117] Anahita Mahzari, Afshin Taghavi Nasrabadi, Aliehsan Samiei, and Ravi Prakash. 2018. Fov-aware edge caching for adaptive 360 video streaming. In *Proceedings of the 26th ACM international conference on Multimedia*. 173–181.
[118] Muhammad Faran Majeed, Syed Hassan Ahmed, Siraj Muhammad, Houbing Song, and Danda B Rawat. 2017. Multimedia streaming in information-centric networking: A survey and future perspectives. *Computer Networks* 125 (2017), 103–121.

[119] Pantelis Maniotis and Nikolaos Thomos. 2021. Tile-based edge caching for 360° live video streaming. *IEEE Transactions on Circuits and Systems for Video Technology* (Feb. 2021).

[120] IBM Watson Media. 1998–2022. The Future of Video with Watson. Retrieved Jan. 11, 2022 from https://video.ibm.com

[121] Wowza media systems. 2007–2022. If You Can Dream It, Wowza Can Stream It. Retrieved Jan. 11, 2022 from https://www.wowza.com/

[122] Rosangela Melo, Maria Clara Bezerra, Jamilson Dantas, Rubens Matos, Ivanildo José de Melo Filho, Aline Santana Oliveira, Fábio Denilson de Oliveira Feliciano, and Paulo Romero Martins Maciel. 2017. Sensitivity analysis techniques applied in cloud computing environments. In *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, Lisbon, 1–7.

[123] Rosangela Melo, Maria Clara Bezerra, Jamilson Dantas, Rubens Matos, Ivanildo Melo, and Paulo Maciel. 2014. Sensitivity analysis of availability of video streaming service in cloud computing. In *2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*. IEEE, Austin, TX, USA, 1–2.

[124] Microsoft. 2009. *Smooth Streaming Technical Overview*. https://docs.microsoft.com/en-us/iis/media/on-demand-smooth-streaming/smooth-streaming-technical-overview

[125] Microsoft. 2020. Protected Interoperable File Format. Retrieved Jan. 11, 2022 from https://docs.microsoft.com/en-us/iis/media/smooth-streaming/protected-interoperable-file-format

[126] Microsoft. 2020. Smooth Streaming Protocol. Retrieved Jan. 11, 2022 from https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-sstr/8383f27f-7efe-4c60-832a-387274457251

[127] Microsoft. 2021. Protect your content with Media Services dynamic encryption. Retrieved Jan. 11, 2022 from https://docs.microsoft.com/en-us/azure/media-services/latest/drm-content-protection-concept

[128] Konstantin Miller, Emanuele Quacchio, Gianluca Gennari, and Adam Wolisz. 2012. Adaptation algorithm for adaptive streaming over HTTP. In *2012 19th international packet video workshop (PV)*. IEEE, Munich-Garching, Germany, 173–178.

[129] Anish Mishra, Sagar Ganiga, Meit Maheshwari, Shreya Saha, and Gaurav Kumar. 2019. Secure and Decentralized Live Streaming using Blockchain and IPFS. In *Third Workshop on Blockchain Technologies and its Applications*.

[130] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1928–1937.

[131] Ahmed Afif Monrat, Olov Schelén, and Karl Andersson. 2019. A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access* 7 (Aug. 2019), 117134–117151.

[132] Muvi. 2022. The global OTT Video & Audio Streaming. Retrieved Jan. 11, 2022 from https://www.muvi.com/

[133] Shahid Nabi, Muhammad Umar Farooq, and Farhan Hussain. 2019. SHANZ Algorithm for QoE Enhancement of HTTP Based Adaptive Video Streaming. In *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*. IEEE, Chongqing, China, 393–400.

[134] Koichi Nihei, Hiroshi Yoshida, Natsuki Kai, Kozo Satoda, and Keiichi Chono. 2018. Adaptive bitrate control of scalable video for live video streaming on best-effort network. In *2018 IEEE Global Communications Conference (GLOBECOM)*. Abu Dhabi, United Arab Emirates, 1–7.

[135] Panopto. 2022. Record, Share, and Manage Videos Securely. Retrieved Jan. 11, 2022 from https://www.panopto.com/kr/

[136] JW player. 2007-2021. We're passionate about video innovation. Retrieved Jan. 11, 2022 from https://www.jwplayer.com/

[137] VLC Media Player. 2022. VLC Media Player. Retrieved Jan. 11, 2022 from https://www.videolan.org/vlc/index.en_GB.html

[138] Shiva Raj Pokhrel and Surjit Singh. 2021. Compound TCP Performance for Industry 4.0 WiFi: A Cognitive Federated Learning Approach. *IEEE Transactions on Industrial Informatics* 17, 3 (Mar. 2021), 2143–2151.

[139] Konstantinos Poularakis, George Iosifidis, Antonios Argyriou, Iordanis Koutsopoulos, and Leandros Tassiulas. 2019. Distributed caching algorithms in the realm of layered video streaming. *IEEE Transactions on Mobile Computing* 18, 4 (Apr. 2019), 757–770.

[140] Qifan Pu, Haoyuan Li, Matei Zaharia, Ali Ghodsi, and Ion Stoica. 2016. FairRide: Near-optimal, fair cache sharing. In *13th {USENIX} Symposium on Networked Systems Design and Implementation (NSDI 16)*. Santa Clara, CA, 393–406.

[141] MA Rajan, Ashley Varghese, N Narendra, Meena Singh, VL Shivraj, Girish Chandra, and P Balamuralidhar. 2016. Security and privacy for real time video streaming using hierarchical inner product encryption based publish-subscribe architecture. In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, Crans-Montana, Switzerland, 373–380.

[142] Farhad Raufmehr, Mohammad Reza Salehi, and Ebrahim Abiri. 2020. A frame-level MLP-based bit-rate controller for real-time video transmission using VVC standard. *Journal of Real-Time Image Processing* (Sep. 2020), 1–13.

[143] Qingmei Ren, Yong Cui, Wenfei Wu, Changfeng Chen, Yuchi Chen, Jiangchuan Liu, and Hongyi Huang. 2018. Improving quality of experience for mobile broadcasters in personalized live video streaming. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, Banff, AB, Canada, 1–6.

[144] Giovanni Rigazzi, Jani-Pekka Kainulainen, Charles Turyagyenda, Alain Mourad, and Jaehyun Ahn. 2019. An edge and fog computing platform for effective deployment of 360 video applications. In *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*. IEEE, Marrakech, Morocco, 1–6.

[145] Eun-Seok Ryu and SunJung Ryu. 2017. Robust real-time UHD video streaming system using scalable high efficiency video coding. *Multimedia Tools and Applications* 76, 23 (May 2017), 25511–25527.

[146] Nouha Samet, Asma Ben Letaifa, Mohamed Hamdi, and Sami Tabbane. 2017. Energy consumption comparison for mobile video streaming encryption algorithm. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, Valencia, 1350–1355.

[147] Yusuf Sani, Andreas Mauthe, and Christopher Edwards. 2017. Adaptive bitrate selection: A survey. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2985–3014.

[148] Michael Seufert, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hoßfeld, and Phuoc Tran-Gia. 2014. A survey on quality of experience of HTTP adaptive streaming. *IEEE Communications Surveys & Tutorials* 17, 1 (2014), 469–492.

[149] Wella Edli Shabrina, Dodi Wisaksono Sudiharto, Endro Ariyanto, and Muhammad Al Makky. 2020. The QoS improvement using CDN for live video streaming with HLS. In *2020 International Conference on Smart Technology and Applications (ICoSTA)*. IEEE, Surabaya, Indonesia, 1–5.

[150] Karthikeyan Shanmugam, Negin Golrezaei, Alexandros G Dimakis, Andreas F Molisch, and Giuseppe Caire. 2013. FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers. *IEEE Transactions on Information Theory* 59, 12 (Sep 2013), 8402–8413.

[151] Yongtao Shuai and Thorsten Herfet. 2018. Towards reduced latency in adaptive live streaming. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. Las Vegas, NV, USA, 1–4.

[152] A. Soltanian, F. Belqasmi, S. Yangui, M. A. Salahuddin, R. Glitho, and H. Elbiaze. 2018. A Cloud-Based Architecture for Multimedia Conferencing Service Provisioning. *IEEE Access* 6 (Jan. 2018), 9792–9806.

[153] Abbas Soltanian, Diala Naboulsi, Roch Glitho, and Halima Elbiaze. 2019. Resource Allocation Mechanism for Media Handling Services in Cloud Multimedia Conferencing. *IEEE Journal on Selected Areas in Communications* 37, 5 (May 2019), 1167–1181. https://doi.org/10.1109/JSAC.2019.2906806

[154] SproutSocial. 2021. *20 Facebook stats to guide your 2021 Facebook strategy*. https://sproutsocial.com/insights/facebook-stats-for-marketers

[155] StreamShark. 2022. Live Stream With Confidence. Make your next live stream a success with StreamShark! Retrieved Jan. 11, 2022 from https://streamshark.io/

[156] Satish Kumar Suman, Aniket Dhok, and Swapnil Bhole. 2020. DNNStream: Deep-learning based Content Adaptive Real-time Streaming. In *2020 International Conference on Signal Processing and Communications (SPCOM)*. IEEE, Bangalore, India, 1–5.

[157] Carlos A Talay, Franco A Trinidad, Diego R Rodríguez Herlein, M Luz Almada, Claudia N González, and Luis A Marrone. 2018. Analysis of the performance of TCP Vegas and its relationship with alpha and beta parameters in a wireless links network and burst errors. In *2018 Congreso Argentino de Ciencias de la Informática y Desarrollos de Investigación (CACIDI)*. IEEE, Buenos Aires, Argentina, 1–6.

[158] THEOPlayer. 2022. THEOPlayer: Universal Video Player. Retrieved Jan. 11, 2022 from https://www.theoplayer.com/

[159] Zhao Tian, Laiping Zhao, Lihai Nie, Peiqi Chen, and Shuyu Chen. 2019. Deeplive: QoE Optimization for Live Video Streaming through Deep Reinforcement Learning. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, Tianjin, China, 827–831.

[160] Tiktok. 2022. Short Live Video Streaming. Retrieved Jan. 11, 2022 from https://www.tiktok.com/live

[161] Anh-Tien Tran, Nhu-Ngoc Dao, and Sungrae Cho. 2020. Bitrate Adaptation for Video Streaming Services in Edge Caching Systems. *IEEE Access* 8 (2020), 135844–135852.

[162] Anh-Tien Tran, Demeke Shumeye Lakew, The-Vi Nguyen, Van-Dat Tuong, Thanh Phung Truong, Nhu-Ngoc Dao, and Sungrae Cho. 2021. Hit Ratio and Latency Optimization for Caching Systems: A Survey. In *2021 International Conference on Information Networking (ICOIN)*. IEEE, Jeju Island, Korea (South), 577–581.

[163] Twitch. 2022. Esport Live Streaming. Retrieved Jan. 11, 2022 from https://www.twitch.tv/

[164] Video Quality Experts Group (VQEG). 2022. StreamSim. Retrieved Jan. 11, 2022 from https://vqeg.github.io/software-tools/encoding/streaming/streamsim/

[165] Video Quality Experts Group (VQEG). 2022. Video Quality Expert Group - Motivation, Objectives and Rules. Retrieved Jan. 11, 2022 from https://www.its.bldrdoc.gov/vqeg/about-vqeg.aspx

[166] World Wide Web Consortium (W3C). 2022. WebIDL. Retrieved Jan. 11, 2022 from https://www.w3.org/TR/WebIDL/

[167] F. Wang, C. Zhang, F. Wang, J. Liu, Y. Zhu, H. Pang, and L. Sun. 2020. DeepCast: Towards Personalized QoE for Edge-Assisted Crowdcast With Deep Reinforcement Learning. *IEEE/ACM Transactions on Networking* 28, 3 (Jun. 2020), 1255–1268.

[168] Junjue Wang, Brandon Amos, Anupam Das, Padmanabhan Pillai, Norman Sadeh, and Mahadev Satyanarayanan. 2018. Enabling live video analytics with a scalable and privacy-aware framework. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 3s (Jun. 2018), 1–24.

[169] Junjue Wang, Ziqiang Feng, Zhuo Chen, Shilpa George, Mihir Bala, Padmanabhan Pillai, Shao-Wen Yang, and Mahadev Satyanarayanan. 2018. Bandwidth-efficient live video analytics for drones via edge computing. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, Seattle, WA, USA, 159–173.

[170] Mu Wang, Changqiao Xu, Shijie Jia, and Gabriel-Miro Muntean. 2018. Video streaming distribution over mobile Internet: A survey. *Frontiers of Computer Science* 12, 6 (2018), 1039–1059.

[171] Ziyi Wang, Yong Cui, Xiaoyu Hu, Xin Wang, Wei Tsang Ooi, and Yi Li. 2020. MultiLive: Adaptive Bitrate Control for Low-delay Multi-party Interactive Live Streaming. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. Toronto, ON, Canada, 1093–1102.

[172] Yongpeng Wu, Xiqi Gao, Shidong Zhou, Wei Yang, Yury Polyanskiy, and Giuseppe Caire. 2020. Massive access for future wireless communication systems. *IEEE Wireless Communications* 27, 4 (2020), 148–156.

[173] S. Xu and G. Liu. 2020. Multi-access Edge Computing Based User Experience Driven Multicast Video Conference Algorithm. In *2020 IEEE International Conference on Edge Computing (EDGE)*. Beijing, China, 99–105.

[174] Xiaodong Xu, Jiaxiang Liu, and Xiaofeng Tao. 2017. Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation. *IEEE Access* 5 (Aug. 2017), 16406–16415.

[175] Zichuan Xu, Weifa Liang, Meitian Huang, Mike Jia, Song Guo, and Alex Galis. 2018. Efficient NFV-enabled multicasting in SDNs. *IEEE Transactions on Communications* 67, 3 (Mar. 2018), 2052–2070.

[176] Jian Yang, Enzhong Yang, Yongyi Ran, Yifeng Bi, and Jun Wang. 2018. Controllable multicast for adaptive scalable video streaming in software-defined networks. *IEEE Transactions on Multimedia* 20, 5 (May 2018), 1260–1274.

[177] Peng Yang, Feng Lyu, Wen Wu, Ning Zhang, Li Yu, and Xuemin Sherman Shen. 2019. Edge coordinated query configuration for low-latency and accurate video analytics. *IEEE Transactions on Industrial Informatics* 16, 7 (Jul. 2019), 4855–4864.

[178] Peng Yang, Ning Zhang, Shan Zhang, Feng Lyu, Li Yu, and Xuemin Shen. 2019. Asymptotic optimal edge resource allocation for video streaming via user preference prediction. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, Shanghai, China, 1–6.

[179] Abid Yaqoob, Ting Bi, and Gabriel-Miro Muntean. 2020. A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities. *IEEE Communications Surveys & Tutorials* 22, 4 (2020), 2801–2838.

[180] Jihyeok Yun, Md Jalil Piran, and Doug Young Suh. Dec. 2018. QoE-driven resource allocation for live video streaming over D2D-underlaid 5G cellular networks. *IEEE Access* 6 (Dec. 2018), 72563–72580.

[181] Kamran Zahoor, Kashif Bilal, Aiman Erbad, and Amr Mohamed. 2020. Service-less video multicast in 5G: Enablers and challenges. *IEEE Network* 34, 3 (2020), 270–276.

[182] Hassan Ibrahim Zawia, Rosilah Hassan, and Dahlila Putri Dahnil. 2018. A survey of medium access mechanisms for providing robust audio video streaming in IEEE 802.11aa standard. *IEEE Access* 6 (2018), 27690–27705.

[183] Ju Zhang, Qian Gao, and Guoqiang Zhang. 2020. Edge Cache Replacement Strategy for SVC-Encoding Tile-Based 360-degree Panoramic Streaming. In *2020 3rd International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, Hefei, China, 122–128.

[184] J. Zhang, Y. Zhang, and M. Shen. 2020. A Distance-Driven Alliance for a P2P Live Video System. *IEEE Transactions on Multimedia* 22, 9 (2020), 2409–2419.

[185] Rong Zhang, Wei Li, Peng Wang, Chenye Guan, Jin Fang, Yuhang Song, Jinhui Yu, Baoquan Chen, Weiwei Xu, and Ruigang Yang. 2020. Autoremover: Automatic object removal for autonomous driving videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 12853–12861.

[186] Xuguang Zhang, Huangda Lin, Mingkai Chen, Bin Kang, and Lei Wang. 2020. MEC-enabled video streaming in device-to-device networks. *IET Communications* 14, 15 (2020), 2453–2461.

[187] Zhao Zhang, Huadong Ma, Yaohong Xue, and Liang Liu. 2017. Fair video caching for named data networking. In *2017 IEEE International Conference on Communications (ICC)*. IEEE, Paris, 1–6.

[188] Zhicai Zhang, Ru Wang, F Richard Yu, Fang Fu, and Qiao Yan. 2019. QoS aware transcoding for live streaming in edge-clouds aided HetNets: An enhanced actor-critic approach. *IEEE Transactions on Vehicular Technology* 68, 11 (Nov. 2019), 11295–11308.

[189] Zhicai Zhang, Ru Wang, F Richard Yu, Fang Fu, Qiao Yan, and Qi Jiao. 2019. QoE Aware Transcoding for Live Streaming in SDN-Based Cloud-Aided HetNets: An Actor-Critic Approach. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, Shanghai, China, 1–6.

[190] Wei Zhao, Wen Qiu, Chuanhua Zhou, Zhi Liu, and Takahiro Hara. 2018. Edge-node assisted live video streaming: A coalition formation game approach. In *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, Abu Dhabi, United Arab Emirates, 1–6.

[191] Chao Zhou, Chia-Wen Lin, Xinggong Zhang, and Zongming Guo. 2019. TFDASH: A fairness, stability, and efficiency aware rate control approach for multiple clients over DASH. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 1 (Jan. 2019), 198–211.

[192] Y. Zhu, Q. He, J. Liu, B. Li, and Y. Hu. 2020. When Crowd Meets Big Video Data: Cloud-Edge Collaborative Transcoding for Personal Livecast. *IEEE Transactions on Network Science and Engineering* 7, 1 (2020), 42–53.