Video Caching, Analytics, and Delivery at the Wireless Edge: A Survey and Future Directions

Behrouz Jedari, Gopika Premsankar, *Member, IEEE*, Gazi Illahi, Mario Di Francesco, *Member, IEEE*, Abbas Mehrabi, *Member, IEEE*, and Antti Ylä-Jääski, *Member, IEEE*

***Abstract*—Future wireless networks will provide high-bandwidth, low-latency, and ultra-reliable Internet connectivity to meet the requirements of different applications, ranging from virtual reality to the Internet of Things. To this aim, edge caching, computing, and communication (edge-C3) have emerged to bring network resources (i.e., bandwidth, storage, and computing) closer to end users. Edge-C3 improves the network resource utilization as well as the quality of experience (QoE) of end users. Recently, several video-oriented mobile applications (e.g., live content sharing, gaming, and augmented reality) have leveraged edge-C3 in diverse scenarios involving video streaming in both the downlink and the uplink. Hence, a large number of recent works have studied the implications of video analysis and streaming through edge-C3. This article presents an in-depth survey on video edge-C3 challenges and state-of-the-art solutions in next-generation wireless and mobile networks. Specifically, it includes: a tutorial on video streaming in mobile networks (e.g., video encoding and adaptive bit-rate streaming); an overview of mobile network architectures, enabling technologies, and applications for video edge-C3; video edge computing and analytics in uplink scenarios (e.g., architectures, analytics, and applications); and video edge caching, computing and communication methods in downlink scenarios (e.g., collaborative, popularity-based, and context-aware). A new taxonomy for video edge-C3 is proposed and the major contributions of recent studies are first highlighted and then systematically compared. Finally, several open problems and key challenges for future research are outlined.**

***Index Terms*—Wireless communications, 5G networks, Internet of Things, mobile edge computing, edge analytics, video analytics, caching, task offloading, video streaming, quality of experience.**

I. Introduction

THE GLOBAL mobile traffic is expected to grow about eight times by the year 2022, where video data will account for about 80% of the traffic [1]. This is not surprising, given that about 60% of the worldwide population has watched videos on their mobile devices in 2018 [2]. In general, videos are generated and distributed by a wide range of user equipment (UE), such as smartphones, smart wearables, or devices in the Internet of Things (IoT). Furthermore, different types of video content are constantly generated in video production (e.g., film and advertisement), augmented reality (AR) applications, and tele-surveillance cameras. Besides, over-the-top service providers (SPs), such as YouTube and Netflix, deliver live video and video-on-demand (VoD) streaming services to their subscribers through websites, mobile applications, or social networks. Indeed, meeting the quality of service (QoS) requirements of video-oriented applications while satisfying user quality of experience (QoE) is very challenging, particularly, due to the time-varying nature of wireless links and UE mobility [3].

As the video traffic over cellular networks grows exponentially, mobile network operators (MNOs) are applying novel technologies in the fifth-generation (5G) of communication networks [26] to meet the QoS/QoE requirements of multimedia applications. The ultimate goal is to deliver high data-rate, low-latency, and reliable multimedia services in enhanced mobile broadband and ultra-reliable low-latency communications [27]. To this end, multi-access edge computing (MEC) [28] has been introduced by integrating cloud computing and wireless networking technologies. The main idea in MEC is to bring computing resources close to end-users within the radio access network (RAN). For instance, deploying *edge servers* at the access points of networks allows MNOs to support applications that require low latency and high-bandwidth video streams. Several commercial MEC platforms have been recently deployed [29], [30], which demonstrates the growing interest in leveraging edge resources to deliver rich multimedia experiences. As a step further, content caching capabilities of information-centric networking (ICN) [31] have been combined with MEC to empower the edge with integrated *edge caching, computing, and communication (edge-C3)* capabilities. In the context of multimedia applications, edge-C3 can simultaneously process and cache video content to provide low-latency and bandwidth-intensive services to users (Fig. 1). At the same time, UEs are also increasingly equipped with more powerful computing and storage capabilities, which allow them to participate in the edge-C3 as well. Moreover, mobile crowdsourcing [32], [33] and device-to-device (D2D) communication [34], [35] enable UEs in close proximity to share their resources with each other, eventually reducing the network congestion and the resources to be used at edge servers. Thus, UEs can also be considered as part of the

Manuscript received April 22, 2020; revised August 23, 2020; accepted September 30, 2020. Date of publication November 9, 2020; date of current version February 24, 2021. This work was supported in part by the Academy of Finland under Grant 299222, Grant 319710, and Grant 332307, and in part by Nokia Center for Advanced Research. *(Corresponding author: Behrouz Jedari.)*

Behrouz Jedari is with the Department of L1 DU, Nokia Corporation, 02610 Espoo, Finland (e-mail: behrouz.jedari@nokia.com).

Gopika Premsankar, Gazi Illahi, Mario Di Francesco, and Antti Ylä-Jääski are with the Department of Computer Science, Aalto University, 02150 Espoo, Finland (e-mail: gopika.premsankar@aalto.fi; gazi.illahi@aalto.fi; mario.di.francesco@aalto.fi; antti.yla-jaaski@aalto.fi).

Abbas Mehrabi is with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K. (e-mail: abbas.mehrabidavoodabadi@northumbria.ac.uk).

Digital Object Identifier 10.1109/COMST.2020.3035427

TABLE I
SUMMARY OF RELATED SURVEYS AND TUTORIALS, SORTED IN CHRONOLOGICAL ORDER (I.E., NEWER LAST). SYMBOLS IN THE LAST FOUR COLUMNS INDICATE THE EXTENT OF CONSIDERATION FOR THE TOPICS IN THE CORRESPONDING HEADINGS: ● FULL, ◐ PARTIAL, OR × NONE

Survey	Main focus	Year	Video	Edge computing	Edge caching	Emerging apps
Pudlewski and Melodia [4]	Compressed sensing and encoding of uplink video data from sensors	2013	●	×	×	×
Seufert *et al.* [5]	QoE in HTTP adaptive video streaming	2015	●	×	×	×
Ioannau and Weber [6]	On-path content caching and delivery in ICNs	2016	×	×	●	×
Shi *et al.* [7]	Edge computing and its use cases	2016	×	●	×	◐
Kua *et al.* [8]	Rate adaptation in HTTP adaptive video streaming	2017	●	×	◐	×
Mao *et al.* [9]	Joint radio/computational resource allocation in MEC	2017	◐	●	●	◐
Mach and Becvar [10]	Architectural aspects of computation offloading in MEC	2017	×	●	×	×
Wang *et al.* [11]	Architectural features, caching strategies, and applications in edge-C3	2017	◐	●	●	◐
Zhao *et al.* [12]	QoE modeling and assessment for video delivery	2017	●	×	×	×
Bentaleb *et al.* [13]	Rate adaptation in HTTP adaptive video streaming	2018	●	×	◐	◐
Din *et al.* [14]	Cache management strategies and their simulation-based evaluation	2018	×	×	●	×
Li *et al.* [15]	Caching techniques in macro-cellular, D2D, HetNets, and C-RAN	2018	×	×	●	×
Li *et al.* [16]	Architecture, management schemes, and design objectives of MEC systems	2018	◐	●	●	×
Liu *et al.* [17]	Architecture of systems, service models, and applications in MEC	2018	◐	●	◐	◐
Parvez *et al.* [18]	RAN-based caching in 5G networks for low-latency applications	2018	×	◐	●	◐
Paschos *et al.* [19]	Tutorial on caching in future wireless networks	2018	◐	◐	●	×
Porambage *et al.* [20]	Integration of MEC with IoT systems	2018	◐	●	×	●
Vega *et al.* [21]	Machine learning-based QoE prediction in video streaming	2018	●	×	×	×
Wang *et al.* [22]	Frameworks, enabling technologies and challenges in edge-C3	2018	◐	●	●	◐
Yao *et al.* [23]	Caching strategies and content delivery in wireless networks	2019	×	◐	●	◐
Barakabitze *et al.* [24]	QoE management/modeling of multimedia streaming in future networks	2020	●	◐	◐	●
Wang *et al.* [25]	Deep learning for emerging applications in MEC	2020	◐	●	●	◐
Our work	Comprehensive survey of edge-C3 for video applications	2020	●	●	●	●

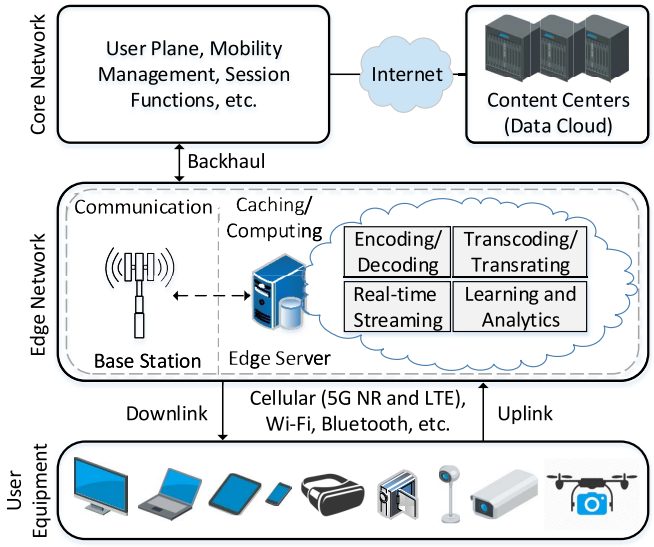

Fig. 1. Abstract view of video edge-C3 in wireless networks.

edge-C3, despite their limited resources compared to edge servers.

Although edge-C3 has been proposed to deliver multimedia-rich applications and services, several challenges remain. First, edge-C3 resources are typically more limited than those available in the cloud data centers. Thus, emerging video-based applications, such as live streaming, AR, and virtual reality (VR), place immense stress on edge-C3 resources. For instance, live streaming applications must simultaneously support low-latency interactions, as well as deliver high-bandwidth data to a large audience. Moreover, in the context of VR, 360° videos demand large storage and bandwidth resources (an order of magnitude higher than traditional video [36]). On the other hand, AR and video surveillance applications must seamlessly process live video frames streamed by UEs to identify and annotate objects in real-time, which requires a large amount of computing resources. Second, the heterogeneity of edge-C3 resources (including edge servers and UEs) raises several challenges on how to efficiently allocate them. Third, the operation and performance of general edge-C3 solutions are significantly affected by the properties of video data (e.g., their encoding models). For instance, caching algorithms for generic content (e.g., in [14]) should be redesigned for segment-based and layered video models to achieve optimal video delivery performance in terms of delivery delay and service cost [37]. Finally, a growing number of UEs (e.g., smartphones, surveillance cameras and mixed reality glasses) generate video content in the *uplink* which requires resource-intensive processing (e.g., to detect objects in a video frame). In this context, careful system design (e.g., efficient placement of encoding services) and allocation of wireless bandwidth for different video qualities are required. Consequently, understanding the properties of video data and their impact on video processing, caching, and transmission performance is extremely important for developing cost-efficient video edge-C3 solutions in wireless networks.

A. Related Surveys and Tutorials

Several existing surveys and tutorials have independently studied the implications of video delivery, edge computing and caching in wireless networks (see Table I). Here, we discuss the most representative publications. First, Mao *et al.* [9] studied joint radio and computational resource management in MEC. They introduced the concept of cache-enabled MEC and

highlighted the benefits of the combined edge-C3 for emerging AR and video streaming applications. Wang *et al.* [11] studied joint edge-C3 resource allocation in wireless networks. However, their study of video applications is mostly restricted to edge caching. Li *et al.* [16] studied the definition of edge computing, architectural features of edge-C3, and resource management therein. They classified the state-of-the-art edge-C3 systems in terms of the objective (e.g., reducing latency, bandwidth, energy). In this context, they considered a few articles related to offloading video analytics tasks to the edge; however, the discussion on video caching is limited. Wang *et al.* [22] studied edge-C3 systems and defined their key performance metrics and frameworks. They discussed a representative AR application (see Section IV-E in [22]) which benefits from edge-C3 for processing both uplink and downlink video streams. However, the authors do not review the state of the art that addresses such use cases. In contrast, we provide a comprehensive review of edge-C3 solutions for emerging multimedia applications, including augmented reality, live streaming, 360° video streaming, and video analytics. Barakabitze *et al.* [24] reviewed QoE management solutions for emerging multimedia applications and edge-based network architectures. Their main focus was on the efficient delivery of video to the users. We consider this aspect as well as the use of edge-C3 resources to efficiently process and deliver videos *generated* by users in the context of live streaming, drone analytics, and video surveillance. Wang *et al.* [25] reviewed deep learning-based applications in edge-C3. In this context, they covered some articles related to video analytics and caching of deep learning results at the edge. However, they mostly considered the machine learning-related aspects of such systems. In contrast, we consider the combined use of edge-C3 resources (caching, computing, and networking) to support analytics applications and are not limited to deep learning-based applications alone. We additionally consider how video-specific characteristics impact the design of edge-C3-based multimedia applications.

To the best of our knowledge, none of the existing surveys specifically addressed edge-C3 in video applications, transmission and delivery. In particular, they have not thoroughly investigated the benefits of both caching and computing for different video applications. The more recent video-centric surveys [21], [24] focus on the QoE aspects of video delivery and adaptation of bitrates for streaming. In contrast, we study the computing, networking, and caching requirements of such applications. The surveys [11], [22] studied the challenges and solutions of joint edge-C3 resource allocation in wireless networks. Nevertheless, they did not consider how the characteristics of video data (e.g., their encoding models, formats, and properties) affect algorithms and protocols in the edge-C3. Moreover, none of them address the benefits of edge computing and caching for emerging applications such as live streaming and 360° video delivery. Furthermore, a study of the use of edge-C3 for video analytics and real-time processing of uplink video data is missing from surveys, except for a deep learning-centric summary in [25]. To fill this gap, this article provides a comprehensive review of video caching, computing, and streaming in wireless edge-C3. Specifically, we study edge-enabled video streaming and analytics in wireless networks for a wide range of emerging applications.

B. Contributions

The primary goal of this survey is to provide the reader with a comprehensive review of the use of edge-C3 for video-based applications. We provide a foundational understanding of video edge-C3 solutions to efficiently process, cache, and stream videos in future wireless networks. Specifically, we focus on edge-C3 solutions to enable emerging applications based on both *downlink* and *uplink* streaming of videos, i.e., wherein UEs consume (e.g., watch) and generate (e.g., record) video data, respectively. To this end, we carefully study high-quality research mainly published since 2012. We provide readers with an in-depth survey of existing edge-C3 solutions, their architectures, and the related challenges. This article mainly targets researchers and practitioners in the fields of telecommunications, network science, computer vision, and data science. Fig. 2 illustrates the organization of the article and Table II summarizes the commonly-used abbreviations.

The main contributions of this article are the following.

- A tutorial on the delivery (streaming) of video over the Internet (Section II). We discuss the core components of video streaming, including encoding, decoding, adaptive streaming, and the related performance metrics. We provide insights into how such streaming solutions can be extended to support emerging applications.
- An insightful overview of networking for video edge-C3 in next-generation wireless and cellular networks (Section III). We overview networking technologies, and the challenges associated with processing and delivering videos both in the uplink and the downlink.
- A thorough review and a new taxonomy of state-of-the-art solutions for wireless video edge-C3. We split the related discussion into two main areas, focusing on *edge intelligence and analytics* for processing video streams in the uplink (Section IV), as well as *edge caching and computing* for efficient delivery of video streams in the downlink (Section V). We carefully review system architectures and optimization problems addressed in these topics, and provide a summary of the lessons learned.
- An overview of open issues and future research directions in wireless video edge-C3 (Section VI). We specifically address selected themes for future work in edge-C3 for video applications and provide a concluding summary (Section VII).

II. Video Streaming Over the Internet: An Overview

We begin with a tutorial on how videos are delivered over the Internet, with a focus on streaming in wireless networks (Fig. 3). We introduce the main components of video streaming (Section II-A), important properties of video data (Section II-B) and types of video (Section II-C). The efficient delivery of videos over a network requires that the videos are converted (i.e., encoded) into different formats. Accordingly, we describe the common encoding standards used today to

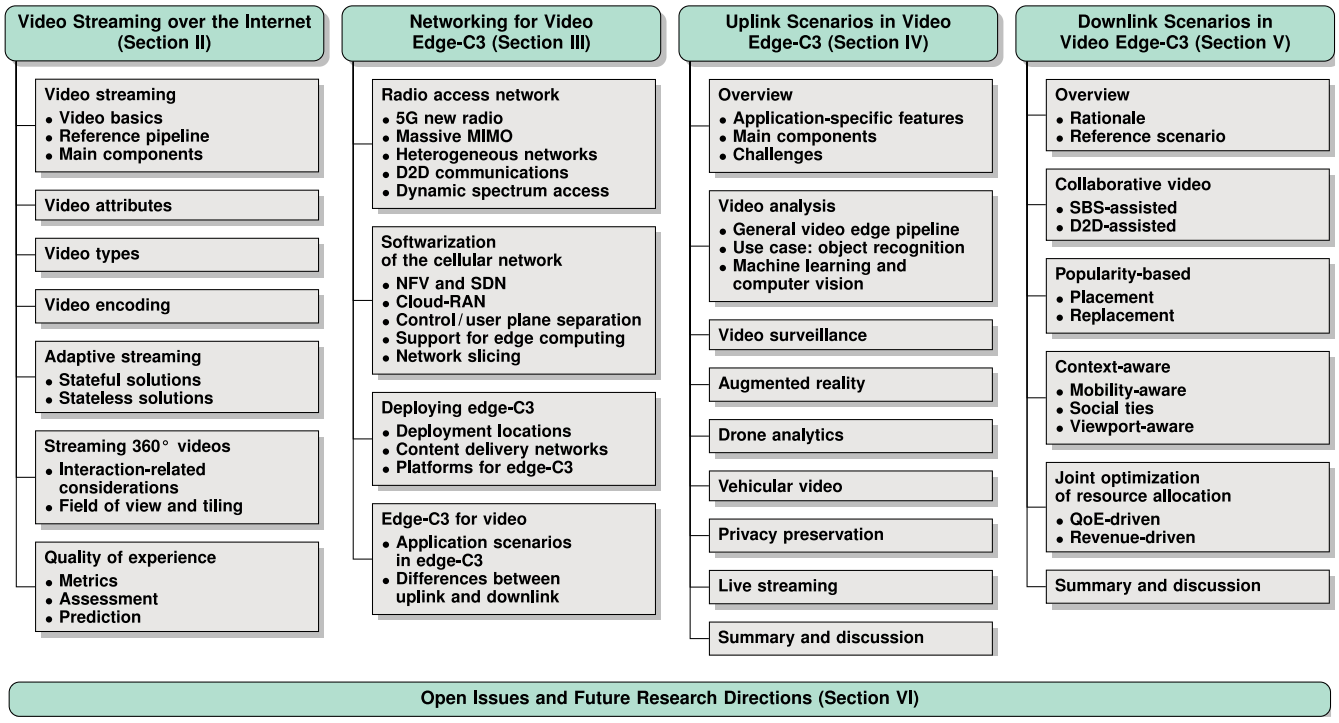

Fig. 2. Organization of the content in the rest of this article.

TABLE II
LIST OF COMMONLY-USED ABBREVIATIONS IN ALPHABETICAL ORDER

Abbr.	Definition	Abbr.	Definition	Abbr.	Definition
3GPP	3rd generation partnership project	IoT	Internet of Things	RTCP	RTP control protocol
AR	Augmented reality	ITU-T	ITU telecomm. standardization	RTP	Real time transport protocol
AVC	Advanced video coding	LFU	Least frequently used	RTSP	Real time streaming protocol
BBU	Baseband unit	LRU	Least recently used	SAND-DASH	Server and network-assisted DASH
BS	Base station	LTE	Long Term Evolution	SBS	Small-cell base station
C-RAN	Cloud radio access networks	MBS	Macro base station	SCTP	Stream control transmission protocol
CDN	Content delivery network	MEC	Multi-access edge computing	SD	Standard definition
CMS	Crowdsourced mobile streaming	MIMO	Multiple-input and multiple-output	SIFT	Scale-invariant feature transform
CNN	Convolutional neural network	MNO	Mobile network operator	SP	Service provider
D2D	Device-to-device	MPD	Media presentation data	SURF	Speeded up robust features
DANE	DASH-aware network element	MPEG	Moving picture experts group	SVC	Scalable video coding
DASH	Dynamic adaptive streaming over HTTP	NFV	Network function virtualization	UE	User equipment
DNN	Deep neural network	ORB	Oriented FAST and rotated BRIEF	UHD	Ultra high definition
Edge-C3	Edge caching, computing and communication	PSNR	Peak signal-to-noise ratio	URI	Uniform resource identifier
FPS	Frames per second	QoE	Quality of experience	VMAF	Video multi-method assessment fusion
HAS	HTTP adaptive streaming	QoS	Quality of service	VoD	Video on demand
HD	High definition	RAN	Radio access network	VQM	Video quality metric
HetNet	Heterogeneous network	RAT	Radio access technology	VR	Virtual reality
HTTP	Hypertext transfer protocol	RRH	Remote radio head	WebRTC	Web real-time communication
ICN	Information-centric networking	RTMP	Real time messaging protocol	YOLO	You only look once

efficiently compress videos (Section II-D). Once the video is encoded, adaptation is still required to ensure that the network can reliably transport the encoded videos even under varying network conditions. We describe adaptive streaming methods to address these issues (Section II-E). Furthermore, emerging video formats (e.g., 360° videos) and VR applications place even more demands on the network due to the large size and format of such videos. To this end, we discuss the streaming solutions proposed for transporting 360° videos (Section II-F). Finally, we discuss performance indicators (in terms of QoS and QoE metrics) that can be used to evaluate video streaming methods (Section II-G).

A. General Video Streaming Pipeline

Video streaming refers to the transmission of an encoded video from one node to another node over the Internet [38]. The two nodes in a video streaming pipeline may be a server and a client or two peers, depending on the architecture of a given video streaming solution [39]. The rest of the discussion assumes a client-server architecture, but the same general

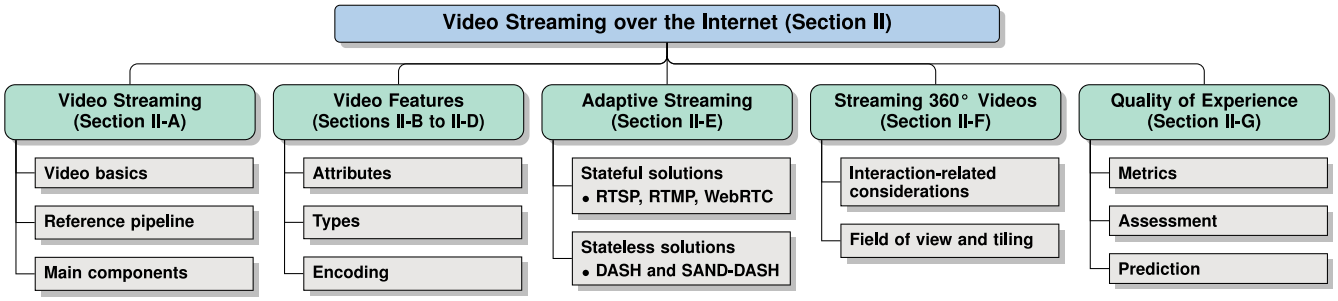

Fig. 3. Organization of the content in Section II.

principles apply to peer-to-peer architectures as well. A key characteristic of video streaming is that the encoded video is progressively downloaded and played out at the same time. As a consequence, the server is required to control transmissions to ensure sustained availability of the video at the client for playout, as opposed to regular file transfer where it is just the completion time that matters [40].

A video is generally captured as a series of still pictures (so-called *frames*) and displayed in rapid succession; the human eyes perceive such as a moving scene [41]. Each video frame is represented as a matrix of individual picture elements (namely, *pixels*). The features (i.e., *attributes*) of the captured video vary (as discussed in Section II-B), for instance, as to compression formats, resolution, and frame rate. Thus, the transmission of a video between devices with heterogeneous capabilities over the Internet poses many challenges, for instance, in terms of transmission delay or used bandwidth [3].

The process of video streaming between a transmitter and a receiver over the Internet can be characterized according to the pipeline in Fig. 4. The main components therein are detailed next.

Video source: Videos can be created in two different ways: as a capture of the physical (i.e., real) environment through a certain device, such as a digital camera, a smartphone, or an IoT video sensor; or as artificially generated (i.e., synthetic) content rendered by a graphic engine. Special use cases, such as AR, may also involve videos in which synthetic elements are overlaid on natural scenes [42].

Encoder: The encoder compresses a source video into a *bitstream* according to a certain format, generally corresponding to a standard (e.g., MPEG-4 AVC). In doing so, the encoder leverages redundant information within the frames to obtain a more space-efficient representation. *Lossless encoding* discards no original information; in contrast, *lossy encoding* may discard some information in the source data. Lossless encoding has a lower compression efficiency than lossy encoding; thus, the latter is widely used in video communications over wireless networks.

Streaming client/server: The streaming server obtains the bitstream and the relevant metadata from the encoder, then repackages the encoded video into a form suitable for transmission over the Internet (particularly, through a *transmission medium*), according to a certain streaming protocol. Such a protocol performs media transport of video *segments* (or chunks) and supports client-server interactions to maintain a

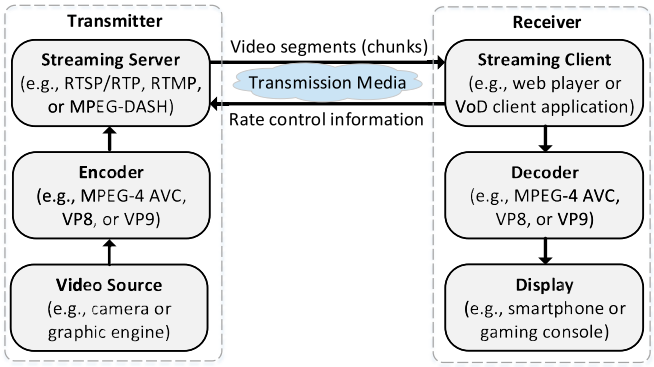

Fig. 4. A general video streaming pipeline over the Internet.

certain level of QoS. The streaming client receives the video bitstream, extracts the encoded video, and feeds it into the decoder. The streaming server or the client may manage the rate adaptation of the streaming session based on dynamic network conditions, depending on the specific use case.

Decoder: The decoder takes the encoded video received by the streaming client and decodes it into its original format. The video is exactly restored into its original form when a lossless scheme is applied; otherwise, the decoded video is (possibly marginally) different from the source. It is worth noting that the quality of a decoded video does not only depend on the encoding scheme, but also on the network conditions (e.g., due to delayed or lost messages).

Transcoder and transrater: Transcoders are widely used in *live* video streaming scenarios. The transcoder decodes a compressed (or encoded) video and re-encodes it with a different scheme (e.g., a different encoding standard or media container). For instance, transcoding is used when streaming clients do not support the video encoding standard of the original video, which requires a conversion to an appropriate format before transmission. In some scenarios, transrating is applied to reduce the bitrate of a video, while keeping the same encoding standard [43]. Both transcoding and transrating improve the scalability of live video streaming by increasing its efficiency and (or) reducing the required bandwidth.

Display: The decoded video is shown on a display device, whose screen comprises a matrix of independent display elements called *display pixels*. The number of display pixels is referred to as the *display resolution*, generally expressed in terms of rows and columns. Display resolutions and sizes vary

from standard to high definition and beyond, due to the diversity of UEs and rich media applications (see Section II-C for a review). There are often differences between the sensor resolution during video capture, the resolution of the encoded video, and the display resolution. Thus, image scaling techniques are employed to make the decoded video fit the display [44].

B. Video Attributes

A video has several features (or attributes) which affect its encoding, transmission, and the resulting QoE. The color information of pixels in a video is represented through a *color space*; for instance, a pixel is defined in terms of red, green, and blue components in the RGB color space. The number of pixels in a video frame is referred to as video *resolution*. In addition, the ratio between the width and height of a video is called *aspect ratio*. The *frame rate* describes the number of frames in one second of a video, usually referred to as frames per second (FPS). The *quality* of a video is its fidelity with respect to the original (uncompressed) version. Quality depends on several factors and can be measured through either subjective and objective measures. The mean opinion score is a subjective measure of video quality obtained from video quality tests involving feedback from human subjects. Subjective video quality testing methodologies in telecommunications have been defined by the ITU telecommunication (ITU-T) standardization sector [45]. Objective measures of video quality operate computationally, for example, by comparing the encoded video against its original (i.e., unencoded) version. Widely used objective video quality metrics include peak signal-to-noise ratio (PSNR), video quality metric (VQM) [46], and video multi-method assessment fusion (VMAF) [47]. The *encoding scheme* is another important attribute of a video, typically including at least the video format (codec), the arrangement of frames, the output FPS, the target bitrate, and rate control. Table III lists the major attributes of videos.

C. Video Types

Emerging applications for video streaming (such as video conferencing, Internet TV, and video blogging) and interactive multimedia [48], [49] (e.g., immersive videos, 3D videos, and mobile AR) employ digital videos of different types. The following categorizes them by application scenarios.

Standard Definition (SD): Refers to videos with a resolution corresponding to that of first-generation digital TV[1] (i.e., 720×480 pixels or 480p). SD videos are commonly employed in VoD and live conversational applications (e.g., Skype and WhatsApp), especially for mobile UEs with comparable screen resolutions.

High Definition (HD): Refers to videos with a resolution corresponding to either high-definition (HD) (i.e., 1280×720 pixels or 720p) or full HD (i.e., 1920×1080 pixels or 1080p) digital TV [50]. (Full) HD videos are commonly employed in VoD and live streaming applications (e.g., sports, cultural events, and game streaming).

[1]For the sake of completeness, the low-definition TV resolutions of 320×240 pixels or 240p and 480×320 pixels or 320p are also employed in the context of wireless video streaming.

TABLE III
THE MAJOR VIDEO ATTRIBUTES AND THEIR DESCRIPTION

Video attribute	Description
Color space	Mathematical model that maps a color representation to its perceptual equivalent
Resolution	The number of pixels in a video frame, expressed as $N \times M$ pixels, where N and M are the number of columns and rows in the pixel matrix (respectively)
Aspect ratio	The ratio of the width to the height of video frames
Frame rate	The number of frames per second (FPS) of a video; higher FPS values translates to smoother visuals
Bitrate	The number of bits needed to represent a second of an encoded video
Video quality	Fidelity of an encoded video to its original version, measured either subjectively or through an objective metric (e.g., PSNR, VQM, and VMAF)
Encoding scheme	A video coding format (i.e., a codec) and relevant encoding parameters

4K: Refers to videos whose width is approximately 4,000 pixels, corresponding to Ultra-HD digital TV (i.e., 3840×2160 pixels) [51]. 4K videos are commonly employed for VoD, IP television, and immersive VR/AR applications – in the latter case, as they need to be displayed very close to the eyes of the viewer.

Multi-view: Describes a scene from multiple points of view to augment the user experience – for instance, to enable 3D tele-immersion applications. The most common form of multi-view is represented by *stereoscopic* videos which are recorded by two synchronized cameras located at the average human inter-pupillary distance. A stereoscopic video is displayed such that each eye can only see the video channel from one of the corresponding cameras, thereby simulating a perception of depth. Stereoscopic videos are mainly used in 3D TV and 3D VR applications.

360°/180°: They are characterized by each frame containing all possible views in every direction so that the whole visual field is captured. Typically, 360° videos are recorded by using multiple synchronized cameras, each capturing a partial view of the observable visual field. The captured views are then stitched together to form the entire observable field. 360° videos are generally used in VR applications; they are also called *immersive* or *omnidirectional* videos [52]. Similarly, 180° videos only capture half of the visual field as a compromise between the level of immersion and ease of production (in terms of capture, processing, and deployment of a video).

D. Video Encoding

Video encoding reduces the redundant information in a video – in both the temporal and spatial domains – through compression. The result is a reduction in the storage size of the video, with minimal (possibly negligible) impact on its quality. Block-based video encoding is a commonly used approach that divides a video frame into multiple rectangles or squares [53]. The size of each block (also called *macroblock*) can vary from 4×4 to 64×64 pixels. If two macroblocks are similar, one can be derived from another. One technique is to *predict* a given macroblock based on those previously encoded as a

TABLE IV
POPULAR CODECS AND THEIR MAJOR FEATURES

Codec	Description
MPEG-AVC / H264 [55]	The most popular video codec, abstracts the network layer from the coding layer
MPEG-HEVC / H265 [56]	Enhancement of MPEG-AVC / H264, improves compression efficiency and parallelization
MPEG-VVC / H266 [57]	Enhancement of HEVC / H265, further improves compression efficiency and supports immersive media (e.g., 360° videos)
VP9 [58]	Royalty-free codec specifically developed for Internet applications, compatible with WebRTC
AV1 [59]	Enhancement of VP9, with better compression efficiency than both VP9 and HEVC

mathematical function. Such a function expresses, for instance, the displacement of a macroblock with respect to the previous one. A frame of predicted macroblocks is subtracted from the actual frame to obtain a residual frame, which is then transformed into a matrix of coefficients (e.g., by applying the discrete Fourier transform). These coefficients are finally quantized according to the specific encoding scheme to obtain a sparse matrix, which reduces the storage size at the cost of some information loss [53].

Several encoding standards have been developed by working groups, such as the ITU-T Video Coding Experts Group and the ISO/IEC JTC1 Moving Picture Experts Group. Additionally, other – generally open-source – encoding formats have been developed by private organizations, such as Google, AOMedia, and Microsoft. Some standards define *network-friendly* encoders that format data and add suitable headers for communication through transport layers over the Internet; they also provide enhanced capabilities to tolerate message errors and losses. Indeed, most video streaming services over the Internet currently use one of these few network-friendly encoders, such as H264, H265, or VP9 [54]. Popular encoders and their major features are listed in Table IV.

1) Scalable Video Coding (SVC): A scalable encoding represents a video as a set of bitstreams (also called *layers*) in such a way that higher quality can be obtained by combining individual (pre-encoded) bitstreams. SVC is the most popular solution in this context, as an extension of H.264/MPEG4 [60] wherein a video includes one base layer and multiple enhancement layers (see Fig. 5(a)). The base layer realizes the first (lowest) quality of the video, the combination of the base layer and the first enhancement layer realizes the second quality of the video, and so on until the highest quality that consists of all layers. Thus, SVC encoding enables flexible video streaming to UEs in wireless networks as it can adapt to fast-varying wireless links without requiring re-encoding [61]. Scalable encoding is particularly beneficial in next-generation wireless networks, wherein streaming servers are located at the edge. In particular, streaming videos with SVC allows to optimize the allocation of edge resources (e.g., caching or computing) to UEs, thereby improving bandwidth utilization and energy consumption [62]–[65]. Tele-conferencing, live Internet broadcasting [66], and video surveillance [67] are common applications of SVC videos.

There are three scalability modes in video coding: spatial, temporal, and quality/fidelity. In the spatial scalability mode, the enhancement layers improve the spatial resolution of a video. For instance, the base layer may provide 480p video, while the combination of the base layer with enhancement layers can increase the spatial resolution to 720p or 1080p. In the temporal scalability, enhancement layers increase the smoothness of a video by increasing its frame rate. For example, the base layer may encode a video at 25 FPS, while the combination of the base layer with enhancement layers can increase the frame rate to 30, 40, or 60 FPS. In the fidelity/quality scalability, the SNR increases with the availability of enhancement layers, while the spatio-temporal resolution of a decoded video is constant irrespective of the number of enhancement layers.

E. Adaptive Streaming

The bitrate of a video is determined by the target quality, depending on the specific codec employed. For adequate QoE, the end-to-end link between the streaming server and the client should have enough capacity to support the transmission rate of the server, namely, it should be at least the same as the source video bitrate. Unfortunately, network conditions generally vary during a streaming session – irrespective from the nature of the communication medium – for different reasons, including congestion, shadowing/fading, and message loss. Sending a video from a server to a client with a constant (bit)rate may either result in poor link utilization if the bitrate is set too low (e.g., as a conservative estimate) or in unsatisfactory QoE due to delayed or lost messages (e.g., choppy or frozen video playout). *Adaptive streaming techniques* have been proposed to address these issues by dynamically adjusting the bitrate of a video according to network conditions.

In general, streaming techniques can be distinguished between *stateful* and *stateless* [68]. Both the sender and receiver store the state of a video streaming session with stateful streaming; whereas only one of the participants may maintain the state of the video streaming session with stateless streaming, thereby releasing the resources of the other participant and allowing scalable operations. Stateful streaming is generally leveraged for live streaming and real-time interactive applications (e.g., cloud gaming), while stateless streaming is commonly employed in VoD applications [54]. The rest of the section introduces commonly-used stateful and stateless protocols for adaptive video streaming.

1) Stateful Adaptive Streaming: This approach employs a variety of protocols; the most representative are detailed next.

The Real Time Streaming Protocol (RTSP) [69] is an application-layer protocol that defines a connectionless streaming session. RTSP leverages two other protocols [70]: the Real-time Transport Protocol (RTP) for end-to-end media transport over UDP; and the RTP Control Protocol (RTCP) to exchange metadata related to the streaming session over TCP, as an out-of-band control and feedback channel. RTSP has a syntax similar to that of HTTP and supports three main operations: retrieving media from a server; inviting a media server to join an existing conference, for instance, to play or record media present therein; notifying a client about the availability

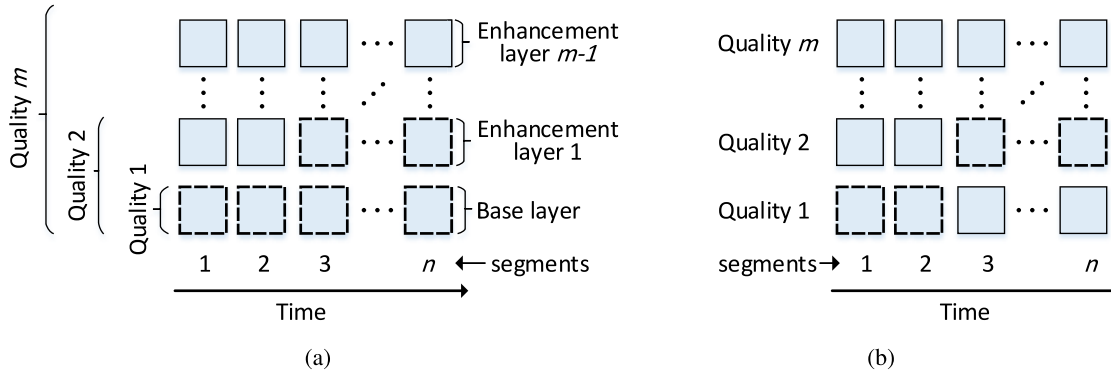

Fig. 5. (a) Scalable and (b) non-scalable representations of a video for DASH streaming. Using the scalable representation, the base layer (i.e., quality 1) is combined with zero up to m–1 enhancement layers to be transmitted as the current segment in each time slot (for example, in Fig. 5(a), quality 1 as segment 2 and the combination of the base layer with enhancement layer 1 as segment 3). Using the non-scalable representation, one quality is selected to be transmitted as the current segment in each time slot (for example, in Fig. 5(b), quality 1 as segment 2 and quality 2 as segment 3).

of additional (new) media, especially useful for live streaming. RTSP supports multicast data delivery.

The Real-Time Messaging Protocol (RTMP) [71] is an application-layer protocol, initially developed as a proprietary solution within the Macromedia Flash multimedia platform; the related specifications are now publicly available. RTMP leverages TCP to maintain a persistent connection between a client and a server, while dynamically splitting streamed data into fragments. The size of fragments is negotiated between the client and server. RTMP maintains multiple parallel channels carrying different data at the same time for efficient and low-latency streaming.

WebRTC is a peer-to-peer protocol for bidirectional exchange of both multimedia and data in real-time between UEs [72]. WebRTC relies on RTP as well as RTCP for media transport and the exchange of control information (respectively); it also supports peer-to-peer data channels through the Stream Control Transmission Protocol (SCTP), a connectionless but reliable transport protocol.

Stateful streaming protocols as those described above are not very suitable for caching, as the streaming session is transient and may not be re-used. However, they can employ transcoding (transrating) to serve video requests of UEs with different wireless link conditions. For instance, a server may transcode an RTMP video stream from a live-streaming client into different qualities, so as to make it available to multiple viewers with diverse link qualities [73].

2) Stateless Adaptive Streaming: A majority of recent stateless streaming protocols use the HTTP protocol for media transmission through so-called HTTP adaptive streaming (HAS), primarily due to the related ease of deployment (through reuse of existing infrastructure) and scalability.

A common feature of HAS protocols is that the streaming server stores multiple representations of a video, each divided into *segments* (equivalently, chunks) that can be independently decoded. The client controls the bitrate of a video by requesting the appropriate segments (generally determined through a local policy) during the streaming process. Adobe HTTP Dynamic Streaming, Apple HTTP Live Streaming, Microsoft Smooth Streaming (MSS), Dynamic Adaptive Streaming over HTTP (DASH) are popular HAS protocols [54]. In the next subsection, we study DASH as the most representative stateless protocol for video streaming.

Dynamic Adaptive Streaming over HTTP (DASH): DASH is a scalable and codec-agnostic HAS protocol developed under the MPEG working group which is supported by telecommunication standardization organizations, such as the 3rd Generation Partnership Project (3GPP) [74]. With DASH, a video is generally represented by multiple *qualities*, where each video quality is divided into multiple segments. Generally, each video segment has a playout length of a few seconds.

Fig. 5 illustrates segmentation in DASH through scalable (i.e., SVC) and non-scalable video representations. In both cases the video is divided into n segments, where each segment is represented according to m qualities. The SVC encoding includes one base layer and m–1 enhancement layers which can be combined to realize m qualities, whereas the non-scalable encoding includes m discrete qualities. A video client can dynamically request video segments with different qualities during a streaming session, since each of them is independently decoded. With DASH, the location of video qualities (in terms of URIs) is stored in a manifest file called media presentation data (MPD). When a DASH client requests a video, the DASH server responds with the MPD file. The client can then start the download of video segments by progressively requesting them from the server, usually through a CDN (see Section III-C for a discussion about CDNs). The decision on the specific quality of each segment at a certain time is realized through a *rate control logic* module at the client side. Such a module evaluates the current network conditions and the buffer occupancy at the client to decide on the appropriate quality [8].

Fig. 6 illustrates a DASH video streaming session in wireless networks.

1) A DASH client in a mobile UE (e.g., a smartphone) requests a video from a DASH server (e.g., by clicking a video link on YouTube).
2) The DASH server sends an MPD file to the client. The client parses the MPD file to obtain information about the quality versions of the video, segmentation

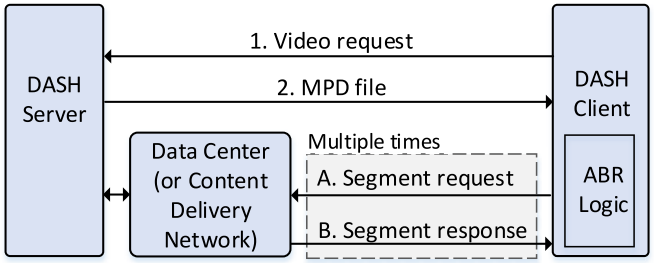

Fig. 6. A DASH video streaming session.

information, and uniform resource identifiers[2] (URI) of all the video segments.

3) The client-side adaptation logic calculates the target bitrate for the next segment, which is then requested through its URI (e.g., from a CDN location).
4) The client downloads the video segment from the server and stores it in its playout buffer.

Steps 3 and 4 are repeated until the last (N) segment of a video stream is received. In conventional DASH deployments, CDNs are networks of data centers while DASH in an edge scenario additionally leverages resources at a base station. Serving segments through a base station clearly reduces both the access time and the backhaul traffic, thereby improving the QoE.

Server and Network-assisted DASH (SAND-DASH): DASH clients generally have limited information about the network conditions. Hence, rate control decisions made by a DASH client might be sub-optimal. Moreover, the service provider has limited control on the QoS of a streaming session, since all the intelligence resides at the client side. In contrast, the DASH server and other nodes in the network – the so-called *network elements* – have a better view of the network status, thus, they can help improve the QoS of the service provider and the QoE in DASH streaming. Accordingly, SAND-DASH [77] has been proposed by introducing DASH-aware network elements (DANEs) that recognize DASH traffic and exchange messages with each other to improve the streaming performance. According to the SAND-DASH architecture, the content server is also considered a DANE and messages may be passed between the DANEs, from DANES to clients, and from clients to DANEs. Messages between DANEs streamline segment delivery and are called Parameters Enhancing Delivery messages; messages from DANEs to clients improve video reception by the client and are called Parameters Enhancing Reception messages; and messages from clients to DANEs may be either status or metric messages. These messages allow both the client and the DANEs to access information relevant for improving DASH performance in terms of both QoS and QoE.

SAND-DASH fits well the architecture of edge-enabled wireless networks, wherein edge resources can be leveraged as DANEs. The collaboration among network entities through message passing enables the design of optimal rate adaptation solutions to jointly improve the QoE of streaming and obtain a fair resource utilization. A number of recent works have studied SAND-DASH in such a context. The authors in [78] design rate-control strategies, such as bandwidth reservation and bitrate guidance, by using edge controllers for SAND-DASH. They also evaluate the proposed streaming schemes in terms of the average video quality received at multiple clients and the fairness in video delivery. Heikkinen [79] proposes an edge-enabled control mechanism for DASH, which employs an optimal slot-based resource allocation policy based on average information on channel state. Experimental results show that the proposed policy reduces the system-wide probability that video playout is interrupted. An edge-enabled rate adaptation system is proposed in [80] through a greedy client/server mapping strategy to jointly maximize the QoE and fairness of competing mobile video streaming clients. The experiments therein demonstrate that the proposed solution outperforms client-based rate adaptation heuristics.

[2]The MPD file may contain direct addresses of the segments in the CDN [75], or the server may employ DNS load balancing to redirect the request to the appropriate CDN location [76].

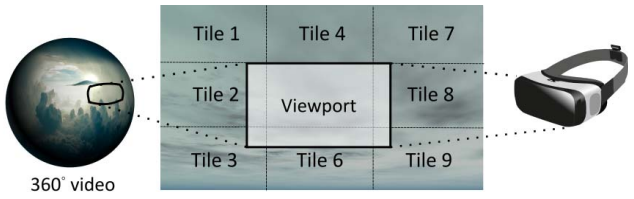

Fig. 7. Division of a 360° video into tiles.

F. Streaming 360° videos

Streaming 360° (or panoramic) videos is more challenging than streaming traditional video content. First, 360° videos require higher bandwidth and storage space than regular videos. Next, such videos allow more interaction, i.e., users can turn their heads as they want and observe different parts of the panorama. Thus, the latency requirements for streaming are stricter as a large delay in updating the display (once a user changes his/her field-of-view) may result in motion sickness [36]. This delay is referred to as *motion-to-photon* latency and must be in the range of a few tens of milliseconds for a smooth viewing experience [36]. Additionally, a streaming solution must support different viewer devices (e.g., head-mounted displays and smartphones) and different wireless network conditions. To this end, adaptive streaming is used for streaming 360° videos as well, with some modifications that leverage the properties of such content. Specifically, the panorama to be streamed is spatially divided into several *tiles* (see Fig. 7), each of which can be encoded into different bitrates. A streaming server can use a tile-based approach to reduce bandwidth by only transmitting the tiles that are visible in the user's *viewport* (i.e., field of view), or transmitting the tiles in the viewport with a higher quality than other tiles. However, such an approach requires that the view is updated with a low motion-to-photon latency when the user's viewport changes. To this end, some articles focus on predicting the user's viewport [81], [82] and accordingly pre-fetching the tiles in the predicted viewport. However, the rendering of the viewport with multiple independently-encoded tiles may require multiple decoders on the viewer's

device. Qian *et al.* [81] address this problem by designing a solution that makes decoding and playback asynchronous. Specifically, they design a decoding scheduler that assigns tiles to idle decoders; the decoded tiles are stored in client buffers, ready to play out when necessary. Finally, the prediction of viewports may be inaccurate, which may result in rendering errors. This can be avoided by the server sending the entire panorama with a low resolution [83] or sending an adaptive number of extra tiles [84] depending on the available bandwidth. The MPEG working group is also in the process of standardizing 360° video delivery and media formats.[3]

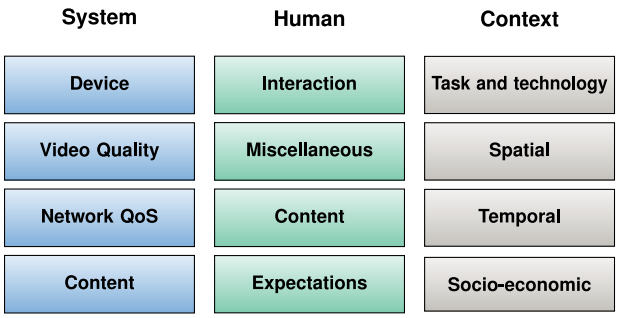

Fig. 8. Main factors affecting QoE in video streaming.

G. QoE in Video Streaming

QoS and QoE are two inter-related but distinct performance metrics. QoS indicates a set of performance metrics that must be fulfilled in delivering a service, even though there is no consensus on its definition [85]. ITU-T defines QoS as "*the totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service*" [86]. In contrast, QoE refers to the actual (i.e., subjective) opinion of users about their experience with a service. Again, ITU-T defines QoE as "*the degree of delight or annoyance of the user of an application or service*" [87]. In the context of multimedia applications, QoE depends on multiple factors, including QoS, the encoding scheme, the quality of content/display, the expectations of the user, as well as contextual parameters (e.g., spatio-temporal or social aspects) [88].

More specifically, the factors affecting QoE in video streaming can be divided into three main classes (Fig. 8): system, context, and human [89]. System factors generally depend on the video attributes such as the viewing device, network QoS, as well as quality and content of the video. System factors often impact on the visual quality and smoothness of the delivered video stream, as perceived by the viewer. Contextual factors include spatio-temporal and socio-economic aspects, as well as those related to the viewing task and the used technology. For instance, QoE can be affected by the location, the time, and the duration of a streaming session. Human factors include user expectations, the level of interaction, and interest in the content. Human factors are viewer-specific, ranging from the emotional and mental state of users to their socio-economic status and even their view of the world.

1) QoE Metrics: Rate control in video streaming affects QoE metrics that describe QoE from a system perspective [5]. The most important QoE metrics are described next.

- *Startup Delay* is defined as the time between the explicit action of a user for watching a video (e.g., a click on the play button) and the time the first segment of the video is played out. Rate control at the client side affects the startup delay, in addition to the network conditions (e.g., the server load).
- *Stalling* occurs when the client playout buffer at a UE becomes empty (also known as buffer starvation) and results in the video becoming "frozen". Stalling mainly occurs due to high server load, network bottlenecks, and non-responsive bitrate adaptation. QoE is affected by both the frequency and the duration of stalls.
- *Bitrate Switching* occurs when the client-side streaming protocol changes the current bitrate to another one (due to adaptive mechanisms), resulting in a sudden change of video quality. QoE is affected by the average quality of the received video, the perception of bitrate adaptation (i.e., how noticeable it is), as well as its frequency.

2) QoE Assessment: Adequate QoE is crucial for both content creators and service providers because it significantly affects customer acquisition, loyalty, and retention. Therefore, maximizing QoE is considered at all stages of video delivery, from network planning to video encoding and rate control in adaptive streaming (e.g., DASH). Clearly, maximizing QoE is not possible unless it can be accurately measured and assessed (see [90] for a survey). A subjective measurement is an ideal benchmark, since QoE varies across different users. However, subjective measurements of QoE are expensive and time-consuming, as they require conducting user studies under very specific viewing conditions. Furthermore, QoE information may be needed in real-time for applications employing adaptive streaming; real-time subjective measurement of QoE is clearly a challenge. As a consequence, QoE is rather *modeled* (i.e., mathematically derived or estimated) as a function of objectively measurable quantities (e.g., using a media player, bitstream, or physical-layer information).

ITU-T has introduced recommendations for non-invasive parametric QoE estimation of audio-visual streaming [91], [92]. These leverage parameters from both the media (e.g., encoder-related) and the transport (e.g., message loss) layers. QoE estimation can be carried out at both the client and at the server in a video streaming pipeline [90]. At the client side, parametric QoE measurement is generally employed in rate adaptation. At the server side, both online and offline QoE measurements are conducted. Online measurements target efficient resource allocation and fairness, improving QoE as a side effect. In contrast, offline QoE measurements are applied to network planning and content management. Juluri *et al.* [90] classify QoE measurement methodologies based on the corresponding data collection approach (e.g., active, passive, or based on user feedback), the place of data collection (e.g., user or network), and QoE metrics (e.g., initial buffer time, stall duration, and re-buffering frequency). The authors in [93] list the factors

[3]https://mpeg.chiariglione.org/standards/mpeg-i

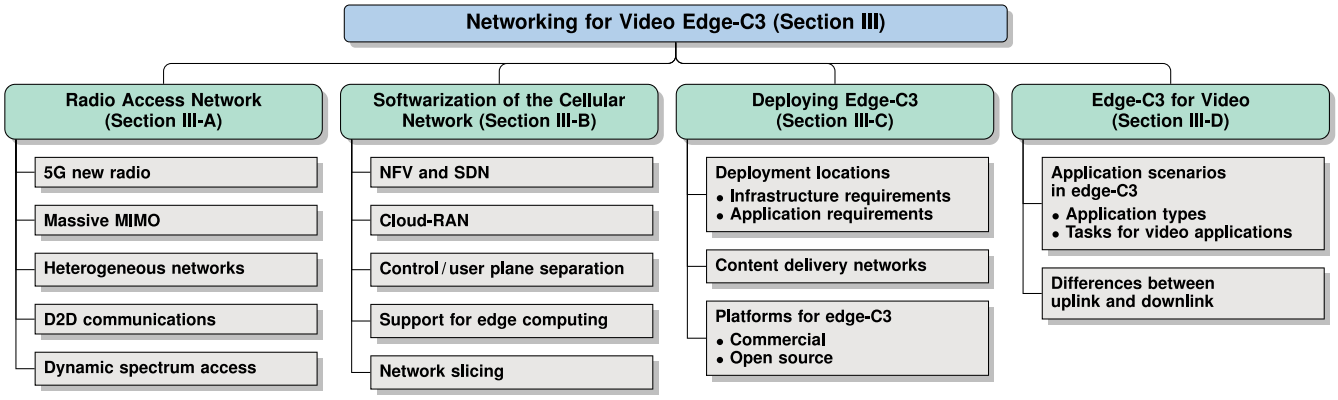

Fig. 9. Organization of the content in Section III.

that affect QoE and classify methods for QoS measurement into subjective and objective.

A QoE model at the network edge can use the resolution of delivered video segments as a measure of QoE [94]. Additional indicators of QoE are represented by the cumulative state of playback in terms of video stalls [95] or effective loss rate over the communication channel [96]. Khan *et al.* [97] propose a QoE model for video streaming which leverages a non-linear function of content type and sender bitrate to predict QoE. However, their proposed model is not validated for adaptive streaming. Nightingale *et al.* [98] employ flow-based QoS metrics in a virtualized environment to model QoE for UHD video streams. In particular, QoE is modeled as a function of content-dependent network parameters. However, the model therein is designed for RTP-based streams and is not suitable for adaptive streaming applications such as VoD. Ge and Wang [99] propose a virtualized, real-time QoE monitoring network function for MEC which utilizes HTTP proxying and packet sniffing to estimate buffer occupancy, quality switching, stalling, and initial playout delay. However, the mapping between QoE and metrics related to the video stream is not discussed. A general QoE monitoring framework in 5G networks is presented in [100]; it employs virtual probes to monitor parameters such as radio resource allocation, transport layer metrics, user behavior, and content characteristics. A proof of concept MEC application is proposed in [101] to model QoE as a function of the quality requested by a UE and its standard deviation, as well as the related stall duration.

3) QoE Prediction: QoE estimation aims at deriving QoE metrics based on other parameters, as previously discussed. QoE estimation could be performed either once or continuously over time to obtain an up-to-date characterization of video delivery. In contrast, QoE *prediction* aims at forecasting QoE in the (short-term) future [102].

QoE prediction has been mainly addressed through machine learning algorithms [21]. In general, existing techniques train an online or offline machine learning model with (objective) QoS and (subjective) QoE parameters. Next, they use the trained model to predict QoE in actual deployment scenarios. Singh *et al.* [103] apply random neural networks to implement a QoE-aware video transcoder for H.264/AVC video. Specifically, playout interruptions and encoding quantization are employed to predict QoE. Li *et al.* [104] propose a rate adaptation algorithm to run DASH as a MEC service that dynamically changes MPD files in DASH based on QoE estimation and network conditions measurements. The work includes a proactive strategy that leverages congestion prediction to further improve QoE.

III. Networking for Video Edge-C3

This section overviews the advances in networking technologies that enable edge-C3 for video applications (Fig. 9). First, we describe the most important features of radio access networks (Section III-A) that support video streaming and related applications. Next, we present the softwarization of the cellular network (Section III-B) as a key enabler for flexible deployment of edge-C3. We then discuss the features of edge-C3 deployments, including their potential locations and software platforms (Section III-C). Finally, we characterize video delivery in both the uplink and the downlink (Section III-D).

A. Radio Access Network

Several new technologies have been included in the radio access networks to support emerging video applications as part of the *enhanced mobile broadband* requirements for 5G [105]. Specifically, enhanced mobile broadband encompasses use cases (e.g., 4K videos, live streaming, AR) that require higher data rates and lower latency than current Long Term Evolution (LTE) networks. Moreover, the number of UEs using such multimedia services is only expected to increase. Thus, 5G must simultaneously support a high connection density as well as a high volume of data traffic per unit area [106].

5G new radio: A new radio interface called *5G new radio* has been introduced to flexibly support different requirements (i.e., high data rate, low latency) through changes in the radio physical layer. Specifically, changes have been proposed in the radio waveforms, subcarrier spacing, and frame structure [105]. 5G new radio also supports data transmission in highly directional beams between the base stations and users through *beamforming* [105]. Beamforming is crucial for transmissions in higher frequencies, for instance, *millimeter wave* frequencies beyond 10 GHz [107]. Such frequencies are

expected to be a part of 5G networks as they offer much higher capacities and data rates than the (sub-6 GHz) frequencies used in LTE networks [26]. However, transmissions in higher frequencies incur increased path loss, as well as blockage from walls and objects [107]; in this regard, highly directional transmissions using beamforming are key in providing sufficient coverage. Consequently, 5G new radio supports beamforming at both the physical and the medium access layer. Moreover, it defines a set of beam management operations to align directional data transfer between users and base stations.

Massive multiple-input and multiple-output (MIMO): Massive MIMO enables high throughput applications by using multiple antennas (e.g., at least 64 of them [108]) at both the receiver and the transmitter [109]. The antennas support both horizontal and vertical beams. This allows parallel data transmissions (called layers) on the same time-frequency for each UE, thereby increasing the overall throughput. Furthermore, multi-user MIMO enables simultaneous transmissions on different layers to multiple UEs [105]. Spatial multiplexing allows base stations to increase the overall capacity by several orders of magnitude [26].

Heterogeneous Networks (HetNets): Another method to increase capacity in the radio access network is through the deployment of HetNets [110], as shown in Fig. 10. Specifically, low-power base stations, called small cell base stations (SBSs), are added to the network to supplement the capacity provided by higher-power macro base stations (MBS). SBSs also help to extend connectivity in regions with coverage holes [109]. HetNets are more cost-efficient than deploying additional MBSs, as the latter requires extensive site planning, particularly in dense urban areas [110]. HetNets also encompass networks that seamlessly combine multiple radio access technologies, including macro cells, small cells, and *wireless LANs*; multiple technologies can provide up to twice more capacity than a pure 5G network [111]. However, HetNets require careful planning and coordination policies to reduce interference between diverse cells [109]. Moreover, it is challenging to provide sufficient backhaul capacity for a large number of SBSs to the core network [112], [113]. Although wired connectivity between MBSs and SBSs has been proposed [113], all SBSs cannot be connected through fiber links due to the high costs [112]. Thus, the choice of backhaul connectivity is left to the MNO [113].

Device-to-device (D2D) communications: Adding base stations in a network to increase capacity is an expensive prospect [114]. As an alternative, network coverage and capacity can be improved by allowing UEs in close proximity to establish direct D2D links to communicate and share their resources with each other. Such communication relies on either licensed spectrum in *inband D2D* or unlicensed spectrum in *outband D2D* [115]. Furthermore, the 3GPP standards include support for multi-hop D2D networks that enable network services for UEs that are outside coverage by using nearby UEs as relays [114], [115]. More recently, D2D communications have also been proposed to circumvent the coverage issues with millimeter wave transmissions [114].

Dynamic spectrum access: Spectrum shortage and underutilization of available spectrum remains a challenge for 5G networks [116]. To this end, *cognitive radios* have been proposed, wherein secondary users (i.e., UEs) opportunistically sense and utilize the spectrum whenever it is not occupied by primary users. Cognitive radios can help increase the spectral efficiency and capacity of networks [116], [117], particularly for multimedia and video streaming services [118]. However, opportunistic spectrum sensing is challenging due to fading, shadowing, and potential security issues [119]. Future networks may rely on a spectrum prediction service instead, and accordingly utilize algorithms to efficiently and dynamically share spectrum between multiple users [117]. New spectrum policies have been proposed [119] and already been deployed in the 3.5 GHz band of LTE networks (in USA) through the *citizens broadband radio service* [120]. Specifically, a three-tiered spectrum access policy is defined to protect the *incumbent* primary users (tier-1) from *priority* secondary users with licenses (tier-2) and *generally authorized* users that are unlicensed (tier-3). More recently, the same spectrum sharing system has been announced for 5G networks,[4] which enables the deployment of private 5G networks to support high-bandwidth services (e.g., in large hotels).

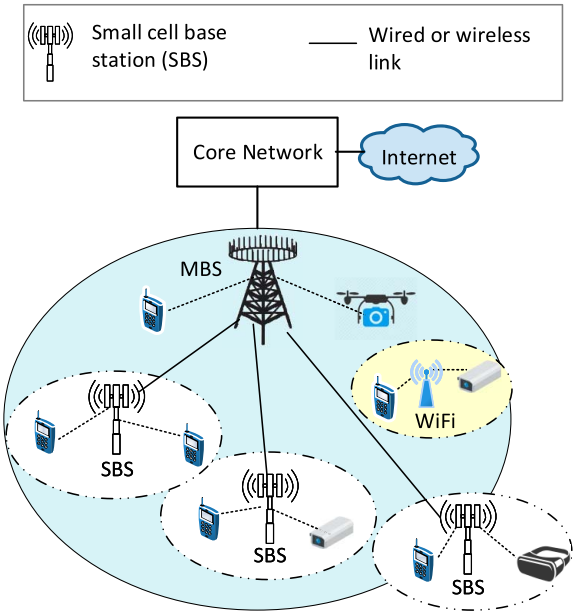

Fig. 10. HetNet architecture.

B. Softwarization of the Cellular Network

The deployment of edge-C3 in a cellular network is enabled by virtualization throughout the network. Specifically, virtualization in both the radio access and core networks plays a key role in supporting flexible deployment of compute and storage resources in different parts of the network.

Network function virtualization (NFV) and software defined networking (SDN): The cellular network is expected to be fully virtualized as part of the NFV [121] paradigm. NFV decouples the network functions from the underlying infrastructure

[4]https://www.cbrsalliance.org/news/cbrs-alliance-opens-gates-for-first-u-s-mid-band-5g-deployments/

to provide flexible deployment of services and network functionality. In particular, the software for specific network functionality (e.g., mobility management) are developed as *virtual network functions* (VNFs) that can run on a standard physical server [121]. Such a virtualized deployment also enables flexible scaling and deployment of functions; for instance, additional instances of a network function can be instantiated on-demand according to the actual traffic. Furthermore, SDN is used to control the flow of data to and from the virtualized functions [122]. SDN, a complementary technology to NFV, decouples the control plane (which makes forwarding decisions) from the data plane (which forwards the data) to provide flexible routing. The control plane functionality is implemented in a logically-centralized controller that can be realized as a software running on general-purpose hardware [121]. Thus, the SDN controller itself may be implemented as a VNF and leverage the benefits of scaling and flexibility offered by such virtualized instances. On the other hand, SDN can benefit NFV by providing the flexible routing required to chain together VNFs to provide services. Thus, SDN and NFV, together, enable the flexible management and programming of the cellular core network.

Cloud radio access network (C-RAN): The architectures of the base stations in the radio access network have also evolved, and thus, can utilize the benefits of virtualization. Specifically, the base station is split into two units – a remote radio head (RRH) and baseband unit (BBU) [109], [123]. RRHs are deployed at base station sites and perform digital processing, analog-digital conversion, power management, and filtering [109]. On the other hand, BBU functions are centralized into *BBU pools* where they can utilize shared, virtualized computing resources to efficiently meet the baseband processing requirements of multiple RRHs. The RRHs are connected to their respective BBU pools through point to point (often optical) links as part of the fronthaul network [123]. Such an architecture (Fig. 11) is referred to as C-RAN [123] and brings the benefits of virtualization to the radio functions. BBU pools are located at more centralized locations, such as the central office of cellular networks [109] or distributed antenna system hubs [124]. In 5G new radio, the BBU functions are further split into distributed units and central units [105], [123]. The lower-layer functions in the networking protocol stack are hosted by the distributed unit, whereas the higher-layer functions are located at the central unit [105]. The C-RAN brings significant savings in capital and operational expenditures for MNOs by relying on centralized and virtualized processing of radio functions [109]. Moreover, the shared processing at BBU pools allows flexible allocation of extra resources when traffic volume is higher [123].

Control and user plane separation: The cellular core network supports mobility, connection establishment, and management of user sessions [122]. The core network relies on control and user (or data) plane separation through distinct functions. Such an architecture also benefits from virtualization. Specifically, the functional split allows the control and data planes to scale independently when deployed as virtualized instances. For instance, content-rich 360° videos and VR scenarios demand a larger volume of data plane traffic. When

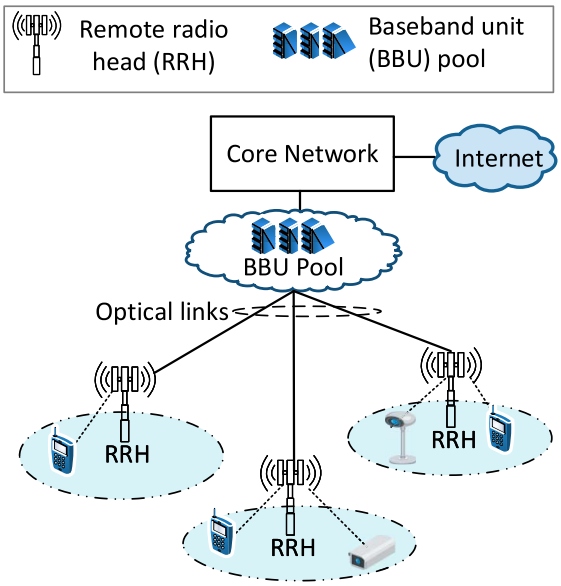

Fig. 11. C-RAN architecture.

the demand for such services increase, the data plane entities may be scaled up to support such demand. Moreover, a fully virtualized environment allows functions to be deployed in locations geographically closer to the users and the traffic to be re-routed accordingly.

Support for edge computing: 5G networks support *service hosting environments* [105] in different locations, where edge computing applications can be deployed as virtualized entities. To this end, the 5G specifications support the flexible deployment of virtualized user plane functions closer to the users to reduce latency. Moreover, decisions about routing are made application-specific and traffic can be steered to a *local area data network* [105], which is geographically closer to the user. Such a local network is accessible by the UEs only from specific locations. Simultaneous access to both a local and centralized data network allows low-latency access to specific applications in the local network [105].

Network slicing: The concept of *network slicing* introduces logical partitioning of the 5G network for different business scenarios or applications [125]. Specifically, a slice comprises a set of network elements specialized in providing a particular type of service [105]. Additional constraints include supporting a certain performance (e.g., latency and data rate) or specific UEs (e.g., corporate customers) [105]. To this end, a new network function – namely, the network slice selection function – has been introduced in the core network to select and create network slices [105]. A virtualized network can be efficiently partitioned on-demand into slices comprising the required network elements and according to the requested QoS [125]. For instance, customized network slices can be created, where each slice is assigned to serve video streaming requests of particular devices (e.g., smartphones, AR glasses, and TVs) with distinct latency and data rate requirements. A slice for 4K streaming may require a caching function, data unit, and cloud unit [125]; whereas a more latency-critical service such as AR may require all functions deployed in the edge.

C. Deploying Edge-C3

The actual location of the caching and computing resources for edge-C3 is not strictly defined. This section describes the potential locations as proposed in the literature, and also presents software platforms that enable the deployment of edge-C3.

Deployment locations: Several locations are proposed by the multi-access edge computing (MEC) specification group (within European Telecommunications Standards Institute) for deploying edge computing in LTE and 5G networks [126], [127]. Specifically, the edge servers may be co-located with base stations, core network functions, or network aggregation points [127]. Examples of network aggregation points include central offices or distributed antenna system hubs where BBU processing is centralized [124]. Choosing a specific location depends on an MNO's technical and business constraints, including the available site facilities and application requirements [127]. Such application requirements include not just latency constraints, but also bandwidth, transport network capacity, and capabilities of the UEs [128]. For instance, co-locating edge servers with base stations results in lowest latency, but incurs a higher deployment cost than at the aggregation points [128]. Moreover, as UEs become more computationally capable, they can be used to carry out some processing themselves [129]. Thus, such devices can be considered as part of the edge as well [10], [130].

Content delivery networks (CDNs): Although the discussion above has discussed computing resources alone, the edge is expected to host both compute and caching resources. Indeed, an alternate location for hosting compute resources is in the network of data centers deployed as part of CDNs used to cache content [128]. For instance, three large CDN providers, namely, Akamai,[5] Cloudflare[6] and Limelight,[7] already support running software functions at the edge. However, these are currently limited to simple functions – with the exception of Limelight, that also allows to run bare metal compute services. The CDNs are usually deployed in points of presence of Internet service providers [131], thus, they are located just outside the cellular network [132]. Recently, proposals have been made to deploy new network functions that reside closer to the users (e.g., co-located with base stations), and obtain radio link information to dynamically select CDNs accordingly [133]. Moreover, local caching at base stations can further reduce the stress on CDNs, for example, during live streaming events [133].

Platforms for edge-C3: Several platforms have been proposed to deploy edge-C3 through either commercial offerings or open-source platforms. Among the commercial solutions, AWS Wavelength [29] enables developers to use the compute and storage resources within the data centers of selected 5G networks. Similarly, Microsoft Azure provides Edge Zones [30] where compute and storage are hosted close to the users, either in data centers of selected 5G MNOs or in private infrastructure on-premise. Both AWS Wavelength and Azure Edge Zones provide a consistent software development experience with realizing and deploying applications on their respective public clouds. However, the support for cellular network providers and locations are currently limited, and may result in vendor lock-in. As an alternative, several open-source platforms have been proposed as well. First, the Linux Foundation Edge[8] aims to build an open and inter-operable framework for edge computing. To this end, Akraino [134] and EdgeXFoundry [135] are the most mature open-source projects within the Linux Foundation Edge. Akraino defines an edge computing platform that supports multiple access network providers, including cellular, wired, WiFi, and IoT networks [134]. It defines a set of application and infrastructure blueprints (i.e., declarative configurations of the entire deployment stack) for different use cases and network deployments. EdgeXFoundry defines an open source software framework that is targeted towards IoT networks [135]. The platform was initially developed to run on IoT gateways, and has since been extended to support both heterogeneous hardware (e.g., gateways, servers and the cloud) and tiered deployments. Next, the Open Networking Foundation[9] is a non-profit, operator-led consortium that includes several projects for transforming the architecture of network providers. Central Office Re-architected as a Data Center (CORD) [136], [137] and Aether [138] are two such projects that target edge deployments. First, CORD utilizes NFV, SDN, and cloud technologies to reconstruct existing infrastructure (e.g., central offices) as data centers [137]. Such an architecture supports flexible deployment of VNFs at the edge to support emerging applications. Aether extends CORD to support an edge cloud-as-a-service platform. Moreover, it supports multiple radio access (licensed, unlicensed, and citizens broadband radio service spectrum), and flexible deployment of VNFs across multiple edge locations.

D. Edge-C3 for Video

The advances in communications and networking of wireless networks highlighted so far enable the high-bandwidth, video-based applications that are the focus of this survey. Moreover, emerging applications rely on processing videos in real-time. This section discusses such application scenarios and highlights the key differences between them.

Application scenarios in edge-C3: Videos can be generated by either the UEs (e.g., smartphones, AR glasses, surveillance cameras) or video content providers (e.g., YouTube, Netflix). Videos published by large content providers are accessed by UEs (i.e., streamed by their subscribers) over the Internet. On the other hand, videos generated by UEs can be either consumed by other UEs (e.g., live streaming), or processed by computer vision algorithms to gain insights from the videos (e.g., live surveillance). Integrating content caching and computing at the network edge can significantly improve the performance of such applications in wireless networks. Specifically, edge-C3, comprising both compute and storage close to the users (at the edge of the network or on the UEs

[5]https://developer.akamai.com/akamai-edgeworkers-overview
[6]https://developers.cloudflare.com/workers/
[7]https://limelight.com/resources/data-sheet/edge-compute/
[8]https://www.lfedge.org/
[9]https://www.opennetworking.org/

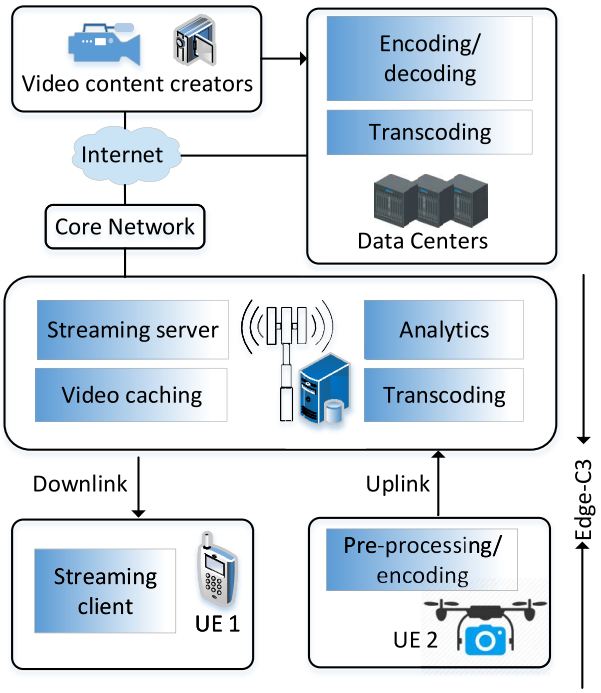

Fig. 12. Video edge-C3 tasks.

themselves), is crucial for efficiently streaming and processing videos. For instance, popular videos can be streamed to UEs from locations closer to the users, thereby maximizing spectral efficiency, improving QoE, and reducing network traffic in the backhaul. Furthermore, edge-C3 can efficiently process videos generated by UEs, extract useful information, and further convert the video streams into appropriate formats that can be served to other users.

Thus, edge-C3 is beneficial for videos generated by UEs (in the **uplink**), as well as videos consumed (watched) by the UEs (in the **downlink**). In particular, some of the important tasks (Fig. 12) that are carried out are as follows. In the downlink, videos from content providers are distributed to viewers using edge-C3 resources that cache and stream videos with a low latency to UEs. On the other hand, in uplink scenarios, videos generated by the UEs are encoded / transcoded in the edge-C3 to efficiently stream such content. Moreover, *analytics* tasks (using either computer vision or machine learning models) are run to derive intelligent insights from the video streams. Certain tasks such as encoding, decoding, and transcoding (see Section II) are required for both uplink and downlink video streams to efficiently support different types of devices and network conditions. However, in the downlink, encoding and transcoding are typically done in large cloud-based data centers before being distributed to UEs.

Differences between uplink and downlink: Uplink and downlink scenarios exhibit different properties which impact resource allocation and application design in edge-C3. First, the wireless bandwidth available in the uplink is typically smaller than that in the downlink. For instance, in 5G networks, the data rates in the uplink are expected to be half of those in the downlink [105]. Thus, interesting trade-offs arise in applications that rely on videos generated from UEs. Such applications must intelligently adapt requirements (e.g., detecting an object within a certain deadline) according to variations in the quality of the video frames (e.g., bitrate, dropped frames) due to constrained uplink bandwidth. Second, the limited computing capabilities in the edge-C3 place constraints on the pre-processing of videos (encoding and transcoding) that are streamed in the uplink. Specifically, streaming videos is demanding in the uplink, as the choice of representations often needs to be made in real-time with limited computing resources. In fact, real-time encoding of 4K videos is not feasible without a powerful CPU or GPU, and sufficient energy capacity [139]. In contrast, in the downlink, content is typically processed offline in to multiple representations (e.g., resolutions and encoding formats to support different devices and network links) on powerful cloud servers and then streamed to users. Third, video content in downlink streaming is typically consumed by human viewers, and thus, adaptation targets improving the viewer's QoE. In contrast, new applications need to run real-time analytics and inference on uplink video streams generated by IoT devices and UEs [140]. Streaming content for such applications is different, as it aims to maximize the quality of the analytics results rather than user-perceived QoE [67], [141]. Finally, applications relying on uplink video streams typically have strict latency constraints (e.g., surveillance, AR, and live streaming) as compared to downlink scenarios (e.g., VoD). Thus, in the uplink, there exist different application-specific considerations than in the downlink for hiding latency from the UEs.

To this end, we classify the works reviewed in this survey into uplink (Section IV) and downlink scenarios (Section V). Specifically, for uplink scenarios, we focus on the processing of videos generated by UEs: how applications can leverage the computing and caching resources in edge-C3 for video analytics and intelligence. On the other hand, for downlink scenarios, we focus on the use of edge-C3 for efficient delivery of videos to the UEs.

IV. Uplink Scenarios in Video edge-C3

This section overviews video edge-C3 for uplink scenarios. In particular, it focuses on emerging applications that leverage video data streamed by UEs (for instance, smartphones, AR glasses, and surveillance cameras). Such applications typically have strict latency requirements for end-to-end transmission and processing, which are highly dependent on the considered use case. For instance, AR demands stringent latency deadlines, whereas live video surveillance places more emphasis on reducing bandwidth of large number of video streams. Accordingly, this section focuses on application-specific approaches at the edge-C3 for processing video streams from UEs (Fig. 13). First, we introduce the main characteristics of applications that rely on streaming videos in the uplink (Section IV-A) and the representative processing tasks in such applications (Section IV-B). Next, we provide a comprehensive review of the state of the art leveraging edge-C3 in emerging applications: live video surveillance, augmented reality, drone analytics, vehicular video analytics, privacy-preserving analytics, and live streaming (Sections IV-C

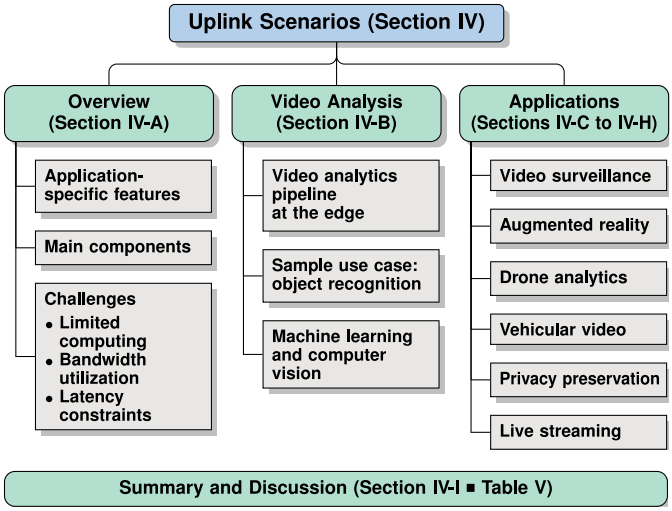

Fig. 13. Organization of the content in Section IV.

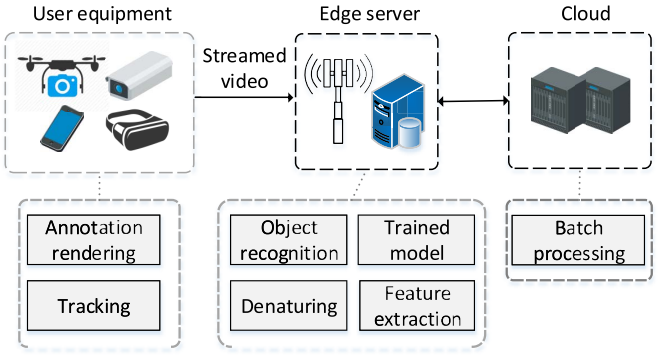

Fig. 14. Main components in video analytics at edge-enabled wireless networks.

to IV-H). Specifically, we take an application-centric approach as many trade-offs are specific to application requirements. To this end, in each sub-section, we first introduce the main features of such applications and describe how the problems of limited bandwidth and computing resources at the edge-C3 have been addressed in the literature. Finally, we conclude with a summary of the lessons learned and highlight commonalities across different applications (Section IV-I).

A. Overview

The edge-C3 allows seamless processing of compute-intensive and delay-sensitive data streamed from diverse UEs such as smartphones, drones, surveillance cameras, and wearables. For instance, surveillance applications can process and query live video streams generated by UEs in real-time. AR is another emerging application wherein video streams from hand-held smartphones or head-mounted displays (e.g., Magic Leap[10] or HoloLens[11]) are processed in real-time so as to overlay useful information for end users. Furthermore, drones and connected cars can take advantage of edge resources for several applications, including surveillance, streaming of sport events, traffic analysis, and parking management. The edge-C3 can also ensure privacy-preserving processing of video streams in different ways. For instance, sensitive information (e.g., faces) can be removed from a video before it is sent to a cloud server for batch processing. Finally, live streaming applications allow normal users to broadcast live video streams from their handheld devices and interact with viewers in real-time.

The applications described above require processing of live video streams to extract information from them and take real-time actions. Computer vision and machine learning models are extensively used to analyze video streams. Thus, relatively powerful computing resources are required at the edge, with multi-core processors of at least 2.7 GHz [142], [143], [144] and powerful GPUs (e.g., in [143], [145], [146]).

[10]https://www.magicleap.com
[11]https://www.microsoft.com/en-us/hololens

However, recent advances in machine learning allow not only edge servers but also resource-constrained UEs to perform complex computer vision tasks. To this end, UEs may carry out less resource-intensive tasks whereas the remaining compute-intensive tasks are *offloaded* to the edge or the cloud. Offloading requires the UEs to transmit video frames (or relevant data such as image features) to the edge-C3 or cloud, where the tasks are run and the results of which are typically sent back to the UE. Fig. 14 provides a high-level overview of an edge-based architecture, along with representative tasks (detailed later) that are carried out at the different layers of the network. It is important to note that the cloud is still required for batch processing of videos, long-term storage, or more resource-intensive computations.

Several challenges remain in the design and deployment of video applications in the edge-C3. First, applications must identify whether analytics tasks (e.g., object detection) are to be processed by the UE itself or offloaded to edge servers (e.g., co-located with base stations). This is challenging when considering the limited and heterogeneous computing capabilities of the edge devices (both UEs and edge servers). Second, the wireless network bandwidth becomes a bottleneck when multiple video streams are streamed to the base stations. Novel approaches are thus required to reduce the bandwidth requirements and allow real-time processing of videos from multiple devices at the edge servers. Finally, latency constraints are very stringent for applications such as AR, wherein the overlay needs to be processed and rendered seamlessly at high frame rates. This requires careful system design to fulfill the requirements of real-time processing applications while meeting their bandwidth and computational constraints.

B. Video Analysis Pipelines at the Edge

Video analytics typically consists of object detection, recognition, and tracking chained together in an *analytics pipeline*. Object detection determines whether an object (face) is present in a video frame or not and localizes the object by drawing a bounding box around the detected object. Recognition consists of object detection and additionally classifying or recognizing its type (e.g., a face). Finally, object tracking in a video requires detecting an object, localizing the object within each video frame, and then tracking the object across frames.

The tasks in an analytics pipeline mainly rely on computer vision algorithms or – more recently – deep neural networks (DNNs) and convolutional neural networks (CNNs). Fig. 15 shows an overview of the main steps in object recognition through computer vision algorithms on edge-C3. Once a video is streamed by a UE, the video is segmented into multiple frames (e.g., images) and the background is removed from each frame in the *pre-processing* stage. Next, *feature descriptors* are extracted from each frame in the feature extraction component. Such descriptors are typically vectors that represent important points in an image (or frame) and are usually invariant, i.e., independent of orientation, scale, or transformation [147]. Some commonly used algorithms to compute feature descriptors are SIFT [148], SURF [149], and ORB [150] for objects, as well as HOG [151] and Haar features [152] for faces.[12] The algorithms differ in terms of the size of the generated vectors and processing time required. For instance, ORB is more efficient and compact followed by SURF and SIFT [147]. In the context of face or human detection, HOG feature descriptors represent the human shape, whereas Haar features describe the appearance (e.g., color and texture) [153]. Once the feature descriptors are extracted, the images can be classified by matching the features through existing models that are already trained with features extracted from a database of images. This step involves using different algorithms, such as nearest neighbor matching or machine learning models [154]. The final results from object recognition can be transferred to the cloud or UEs. In the case of object tracking, feature descriptors are additionally used to track and localize objects in different frames.

CNNs have become very popular in computer vision recently as they do not require feature descriptors to be selected beforehand; instead, the features of an image are automatically learned through the different layers in the CNN. In edge-assisted video analytics, CNNs are typically trained offline against a database of images and then deployed on edge devices to run inferences on the video frames. The popularity of CNNs started growing in 2012, when AlexNet [155], a CNN model, achieved a low top-5 error rate (that measures the presence of the correct label in the top five predicted classes) in classifying images from the ImageNet dataset.[13] This was achieved through the use of GPUs and a deep network architecture [156]. Other noteworthy architectures that appeared thereafter – VGGNet [157] and ResNet [158] – achieved better performance by increasing the depth of the networks [156]. Next, new models emerged that specifically address the problem of object detection (i.e., drawing a bounding box around an object in addition to classifying the object therein). To this end, R-CNN [159] uses a region-based approach, wherein a CNN is used to extract features from 2,000 different region proposals in an image. The features are then fed into a classifier to detect the presence of objects in the proposed regions. However, the computational time and memory required for training R-CNN models is very high [156]. A real-time object detection, YOLO [160], was proposed to address this issue. This model uses a single CNN to predict both the bounding boxes and the class probabilities in each box directly from an image in a single evaluation. Thus, it is able to perform inference in real-time at 45 frames per second [160].

[12]The interested reader may refer to [147] for a review of feature descriptors for objects and [153] for a review of descriptors for humans.

[13]ImageNet (http://www.image-net.org/) is a popular dataset commonly used in computer vision research.

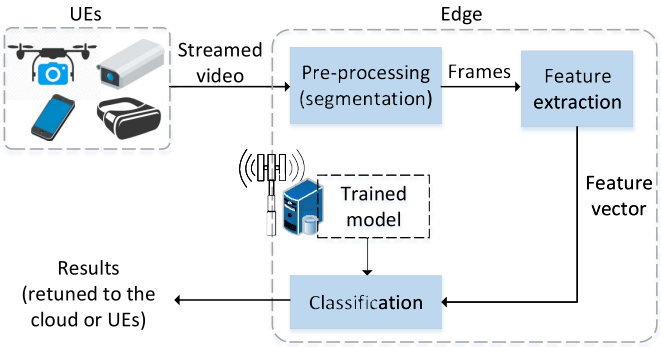

Fig. 15. Object recognition in video streaming at the network edge (adapted from [154]).

Running video analysis and computer vision tasks by resource-constrained UEs or edge servers requires lightweight versions of standard computer vision algorithms and CNN models. For instance, Drolia *et al.* [161] demonstrate how the resources required by SIFT, SURF and ORB descriptors can be reduced by changing the number of extracted features. This results in lower processing time at the expense of a small decrease in accuracy. Similarly, reducing the number of layers in CNNs lowers the storage and computational requirements at the expense of a small decrease in accuracy. Furthermore, layer reduction has the added benefit of lower inference latency. Finally, the CNN models can be specialized for the particular task for which they are intended; for instance, CNN models can be trained to detect objects of a specific color. Such specialized models are smaller in size and require less time for inference. The approaches discussed above have been implemented in different application scenarios, which we discuss next.

C. Video Surveillance

Video surveillance systems involve *queries* (generated by UEs or at cloud servers) to detect or track objects such as humans or cars. Real-time surveillance of videos can help locate a target person (e.g., missing child) or detect dangerous situations such as slippery roads. The analytics pipeline for such systems typically includes object detection, tracking, and recognition. In the context of surveillance, the analytics pipeline is sometimes referred to as a *query plan*, as each query requires a set of analytics tasks to be carried out. We study edge-assisted surveillance applications from the following perspectives: task offloading models, trade-off between latency and accuracy of queries and collaborative processing of surveillance queries.

1) Task Offloading Frameworks: Several recent works have studied strategies for offloading video analytics tasks to edge servers by taking into account the latency, network bandwidth,

and computational constraints. Offloading frameworks need to make complex decisions to choose whether tasks are executed on the UEs, at edge servers, or on the cloud. Specifically, the limited computational resources on the UEs and the edge servers may be stressed when the number of queries increases. Moreover, edge servers may have intermittent connectivity which affects what tasks can be realistically offloaded to such devices. Finally, task offloading decisions need to consider the network bandwidth required to offload videos to edge servers.

The following studies propose task offloading frameworks for video surveillance with different objectives. Trinh *et al.* [142] design an *energy-efficient* task offloading framework in which facial recognition tasks submitted by UEs can be executed by edge servers or the cloud. The offloading decisions take into account the energy and latency requirements, as well as the workloads at edge and cloud servers. In addition, the authors propose an energy-aware routing algorithm for data forwarding that is aware of the network conditions and node failures. Offloading decisions are evaluated through experimental evaluation, whereas the routing algorithm is evaluated through simulations. Li *et al.* [162] focus on the latency and bandwidth constraints of video surveillance. They propose a distributed deep learning approach for object recognition in video streams. Specifically, they optimally place deep learning layers at edge servers with respect to latency and bandwidth constraints. Both online and offline schedulers are proposed to maximize the number of tasks (layers) deployed at the edge. Simulation results show that this solution outperforms other task placement schemes in terms of the number of offloaded tasks with guaranteed QoS. Ding *et al.* [144] focus on the problem of limited radio spectrum and propose a cognitive radio access for data delivery between UEs and edge servers. The placement of tasks take both the limited computation and spectrum resources into account. Caching resources at the edge are employed to temporarily store data when the wireless spectrum is not available. The authors propose a mixed integer linear programming formulation to achieve an optimal task placement that maximizes the number of queries served. The specifics of the tasks from the analytics pipeline are not presented; instead, the computational requirements are modeled as CPU cycles.

2) Accuracy-Computation Trade-Off Analysis: Performing computer vision tasks with very high accuracy is not always the main objective in surveillance applications at the wireless edge. The reason is that achieving very accurate results often requires more edge computing resources as well as higher wireless bandwidth to transmit high-resolution frames. The following articles leverage this trade-off by designing task schedulers for edge servers. Zhang *et al.* [163] empirically characterize the accuracy of computer vision tasks with different settings (e.g., frame resolution, sampling rate) against the resources required to execute them. The data are then used in a query scheduler that places tasks appropriately based on the available resources, required accuracy of results, and a latency threshold. The proposed solution is evaluated over representative datasets (of videos and queries) and found to outperform a fair scheduler by 80% in terms of the quality of results. In contrast to [163], Hung *et al.* [164] characterize the accuracy of different implementations of analytics tasks – including both CNNs and computer vision algorithms – against both resource requirements and network conditions. Moreover, tasks in the analytics pipeline (or query plan) can be reused for different queries. Based on these insights, the authors propose a binary integer program to determine a query plan that maximizes the accuracy of the tasks upon placement on heterogeneous clusters. The objective of the scheduler is to maximize the accuracy of the query result while taking into account the cluster capacity. The proposed formulation has an exponential time complexity and, thus, a greedy heuristic is proposed. The authors evaluate their solution over representative video datasets and find that the accuracy is 5.4 times higher than what is achieved in [163]. Yi *et al.* [165] focus on the trade-off between accuracy and speed, and rely on client-side adjustments of video resolutions to address this. They propose a mixed integer non-linear programming-based scheduler that uses empirical data on the accuracy of analytics tasks for different devices with varying resource capabilities. The scheduler places tasks on edge servers with the objective of minimizing the overall latency while meeting bandwidth constraints. The scheduler is then evaluated through experimental evaluation in terms of number of tasks executed per second as well as the response time per client query. The proposed solution outperforms other baseline algorithms.

3) Cooperative Processing: Some studies incorporate a collaborative approach for processing surveillance queries. In particular, cooperative processing leverages overlapping videos generated by different UEs of the same scene to maximize the accuracy of computer vision tasks. Moreover, edge servers can utilize the spatio-temporal locality of surveillance queries to re-use tasks for different queries. In this context, Lu *et al.* [166] propose a computing platform for cooperative object detection on videos generated by smartphones. The components from the analytics pipeline are performed either on UEs or the cloud. The object detection tasks are carried out with CNNs on CPUs (and not GPUs) of the smartphones. Thus, the processing time becomes shorter when the video frames are processed in batches. Based on this observation, an integer linear programming solution is formulated to determine the optimal number of batches, as well as the number of frames in each batch, such that the computation latency is minimized. Furthermore, a heuristic is proposed to determine the batch features and decide whether to offload the tasks to the cloud. The system is implemented on Android phones and evaluated through experiments. The cooperative processing approach results in a two times speedup with respect to state-of-the-art offloading platforms such as MAUI [167]. Long *et al.* [168] apply cooperative processing on smartphones to detect humans in video streams. In contrast to [166], the offloading framework in this work is not specific to CNNs. The authors propose an integer non-linear programming formulation to partition video analytics tasks, create groups of edge devices, and assign the partitioned tasks to the edge devices. The objective is to maximize the accuracy of detection of all tasks within a latency threshold. The proposed solution is evaluated through simulations and the accuracy is found to be higher than a non-cooperative approach. Zhang *et al.* [169]

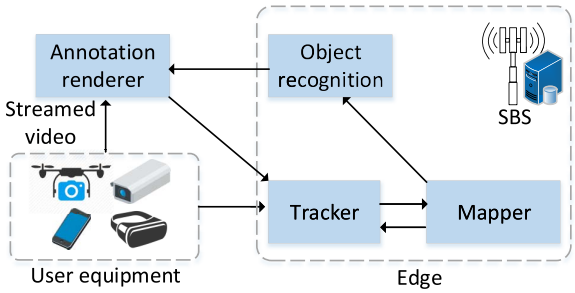

Fig. 16. The analytics pipeline for AR applications (adapted from [171]).

propose a video surveillance system that uses collaborative data from clusters of cameras with overlapping views to improve the accuracy of object detection. Edge servers process the data streams from multiple cameras and transfer analytics information to the cloud for further processing. The system aims to send the maximum number of frames having a queried object. Finally, Jang *et al.* [170] address the collaborative processing of video frames by multiple applications on a surveillance camera. To this end, they design a practical framework that allows multiple virtualized applications to simultaneously access the video stream from a single camera. They then address the configuration of video parameters to support different application-specific QoS requirements.

D. Augmented Reality and Continuous Mobile Vision

Augmented reality (AR) and continuous vision applications display an overlay of virtual information on live videos streamed from devices such as head-mounted displays and smartphones. The AR pipeline (Fig. 16) requires both recognition and tracking of objects detected within video frames [171]. Additionally, a *mapper* builds a model of the environment. Finally, an overlay is rendered on the video shown to end users. The tracker, mapper, and object recognition modules can be offloaded to the edge servers, whereas rendering of the overlay has to be carried out on the UE. The latency requirements in AR applications are very strict. For instance, objects need to be recognized and the overlay rendered before users change their field of view. The overall latency of the entire pipeline needs to be lower than 100 milliseconds [146]. Thus, it is challenging to offload tasks from the analytics pipeline to edge and cloud servers due to the very low latency constraints and limited wireless network bandwidth for video in the uplink. We classify existing studies in AR into task offloading frameworks and trade-off between accuracy and latency of AR tasks.

1) Task Offloading Frameworks: Offloading tasks from an AR pipeline to edge or cloud servers should take into account strict latency constraints, in addition to the wireless bandwidth and computational constraints. Moreover, the limited energy requirements of UEs should be considered when making offloading decisions. To this end, Al-Shuwaili and Simeone [171] study task offloading in *multi-user* AR applications through edge servers with the aim of minimizing the energy consumption of UEs. In particular, the authors propose to share the CPU cycles required by common tasks offloaded by multiple users. Only one user is required to send the data stream to the edge server if multiple users offload similar content or tasks (e.g., image recognition of objects from the same view). The authors propose an optimized task allocation formulation for offloading under such assumptions, which is then numerically evaluated. The works in [146], [172] design edge-assisted AR systems for low-latency object recognition at high frame rates (e.g., 60 FPS). Their main goal is to offload the computationally-heavy recognition tasks to edge servers, whereas the relatively faster tracking algorithms are performed on UEs. Local object tracking allows UEs to render overlays when the output is received from the recognition tasks. Such an approach results in improved detection accuracy with very low resource consumption on UEs. Accordingly, network bandwidth is saved by not sending *all* frames to the edge server or by lowering the encoding quality of uninteresting portions of the frames.

2) Accuracy-Latency Trade-Off Analysis: As latency is a stringent constraint for AR applications, several articles analyze the trade-off between latency and accuracy of the analytics tasks. In particular, a high accuracy is not always required for recognition tasks in AR – a good-enough result is often better than a late but very accurate result. Such a trade-off can be leveraged to determine the placement of analytics tasks for AR applications, which we discuss next. Han *et al.* [173] empirically examine the trade-off between the accuracy of several DNN models and their resource utilization in terms of memory, energy, and latency. Accordingly, the authors propose an algorithm to choose a certain variant of the model and where to execute it. The goal is to maximize the accuracy of the analytics tasks under resource utilization budgets and latency constraints. Similar to [173], Ran *et al.* [143] first empirically characterize the trade-off between accuracy and different attributes of the videos (frame rate, resolution, bitrate). In addition, the authors consider the impact of network conditions (bandwidth and latency) on accuracy. Again, empirical measurements are employed in an optimization problem to choose the most appropriate configuration of the tasks to maximize accuracy. The authors then propose an online heuristic algorithm to achieve a near-optimal result. The proposed solution adaptively configures the settings of the analytics tasks under varying network conditions and achieves a higher accuracy than [173]. Liu *et al.* [174] design a multi-objective optimization problem to optimally assign edge servers and video resolutions to end users. The objective function includes a weight parameter to characterize the accuracy-latency trade-off at different resolutions. Specifically, a high-resolution video can increase the detection accuracy at the expense of longer latency. The proposed formulation is a mixed integer non-linear problem that cannot be solved efficiently. The authors design an algorithm using the block coordinate descent method to find a near-optimal solution. Their solution is evaluated through simulations and a prototype implementation. The latency of the task placement is overall lower than other baseline approaches with minimal loss of accuracy, even under scenarios where the edge server is overloaded or the network latency is significant. Finally, Drolia *et al.* [154], [161] examine the trade-off

between the accuracy and latency of computer vision algorithms. The authors find that the accuracy increases along with an increase in latency as the number of extracted features from an image increases. Thus, the authors propose dynamically adjusting the number of extracted features to minimize latency. Moreover, only relevant parts of the trained computer vision model are stored at the edge based on the spatio-temporal features of the requests. The proposed system reduces the latency of image recognition tasks while maintaining accuracy under different conditions.

E. Drone Video Analytics

Edge-assisted analytics has recently become popular for videos streamed from unmanned aerial vehicles or drones. Drones are increasingly being used in surveillance and mission-critical rescue applications. They represent a different class of surveillance applications due to the different capabilities of drones. For instance, drones are mobile, which affects the decisions on when to offload frames to edge servers. Moreover, the computing resources available on the drone are affected by the form factor and weight of the processing units. Accordingly, we classify the works in this area into those which carry out analytics on the drones themselves and those that offload computation to edge servers.

1) On-Drone Processing: Performing analytics on drones is becoming popular due to the increasing availability of small computing boards that allow running complex computer vision algorithms locally [175], [176]. However, the size and weight of a hardware board attached to a drone impact its flight time and energy consumption [177]. Thus, analytics on-board the drones demand less computationally intensive algorithms and CNN models. Accordingly, Tijtgat *et al.* [176] analytically determine the energy requirements for a drone to carry a certain mass on-board. The authors then evaluate the quality (in terms of the precision and achievable frame rate) of standard CNN models (e.g., YOLO and TinyYOLO) and feature descriptor-based approaches with different video resolutions. The results show that YOLO outperforms other solutions with a higher frame rate and high accuracy. Next, Azimi [175] designs a lightweight CNN model that is specialized for the real-time detection of vehicles. The proposed CNN model is compared with state-of-the-art CNN models; the obtained results demonstrate the feasibility of applying the devised model on drones.

2) Task Offloading: Offloading tasks from drones to edge servers can reduce the computational demand on the drones at the expense of increased use of wireless network bandwidth. To this end, the following two solutions aim to reduce bandwidth requirements of offloading. First, Wang *et al.* [178] investigate methods to reduce network bandwidth requirements by offloading only certain video frames selected using context-aware information (e.g., a specific color in the video frame). The experiments show that applying context-aware filters also significantly reduces the computational requirements of object detection algorithms on edge servers. Next, Chowdhery and Chiang [179] apply edge computing to generate image mosaics from aerial images captured by drones (see Fig. 17). The computationally expensive components of

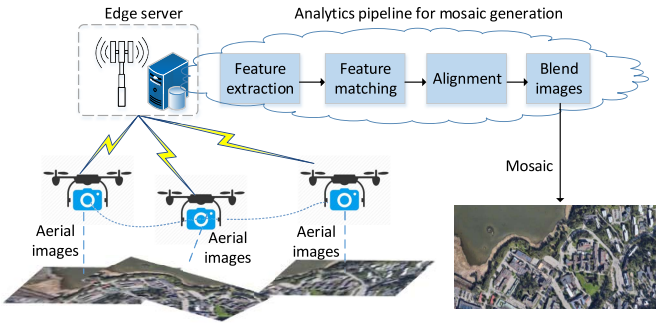

Fig. 17. Overview of generating a mosaic with aerial images captured by drones.

the pipeline, such as feature extraction, are carried out at edge servers. The drones compress captured video frames and offload only selected frames to the edge server to save limited bandwidth. Moreover, the drones run a predictive algorithm to maximize the utility of the application by adjusting the compression parameters based on real-time feedback about the quality of images from the edge server. Different from the above, Wang *et al.* [180] study real-time video streaming at sports stadiums. In particular, video streaming servers are deployed near a stadium to which the drones stream their data over a wireless network. A controller running at the edge server decides the paths of the drones and assigns each drone to a streaming server. The authors propose a joint optimization problem to maximize both the coverage and quality of video streaming. The proposed system is evaluated through simulations and achieves 94% coverage with a high average quality of the video streams.

F. Vehicular Video Analytics

Vehicular analytics typically consist of queries to detect or track objects (e.g., vehicles and parking spaces). Detecting license plates of cars (e.g., using OpenALPR [181]) is also an important component of such analytics applications. The related pipeline consists of first detecting a license plate and then carrying out character recognition on the video frames. The components of this pipeline can be offloaded to the edge servers. However, the mobility of vehicular UEs makes offloading tasks to edge servers extremely challenging. Nevertheless, it is possible to run complex video processing tasks (e.g., object detection) on vehicles, as they today have sufficient computational capabilities. To this end, the following two articles present frameworks for vehicular data analytics. Zhang *et al.* [182] present an open-source vehicular data analytics platform – namely, OpenVDAP – which distributes computing tasks of the analytics pipeline over multiple vehicles and edge servers. Additionally, OpenVDAP addresses the sharing of data between different applications deployed at the wireless edge. Zhang *et al.* [183] present an edge analytics framework called Firework. Firework allows sharing of data from multiple sources for different applications. The authors focus on developing a programming interface that allows software developers to program applications on top of the proposed framework. The authors evaluate their solution with an application to detect license plates. The following systems leverage other mobile devices such as smartphones to

carry out analytics. Qiu *et al.* [184] design a system to track a car's path over a network of fixed surveillance cameras that uses computational resources on mobile devices (e.g., smartphones and cameras on-board vehicles) when necessary. Specifically, the tracking system uses a light-weight analytics pipeline on the mobile devices. In addition, it uses a resource-intensive pipeline on the cloud, consisting of object detection, tracking, and association of cars between video frames captured by multiple cameras. Processing is carried out on mobile UEs only when the results from the cloud have low confidence. Finally, Grassi *et al.* [185] present a system to detect vacant parking spots in a city using video streams captured by smartphones. Analytics are carried out on the smartphones and the output is sent to the cloud, where data from multiple cars is aggregated.

G. Privacy-Preserving Analytics

Applications deployed at edge servers can enhance the privacy of users by removing sensitive information from their videos before sending them to the cloud. This concept was first introduced in a system for analytics on crowd-sourced videos [186]. The system allows users to specify privacy policies (e.g., to blur faces) that are applied to their streamed videos. The policy is implemented through a *denaturing* process. The denaturing pipeline consists of first detecting an object (e.g., a face), recognizing the object to apply user-specific policies, and finally applying a blur or filter to the region. In this pipeline, interesting trade-offs arise between the achieved throughput, accuracy of denaturing, and different video resolutions. The denaturing process is further explored in [187], where the authors extend the denaturing pipeline to speed up the overall process. In particular, a tracking component is added to track already recognized faces across video frames to prevent multiple invocations of the recognition algorithm. Fig. 18 presents the improved denaturing pipeline with added revalidation tasks to prevent drifting of the detected boxes (around the object or face) as well as the option to save encrypted original frames. Finally, the authors describe a policy for reversing the denaturing process by trusted third parties (such as the police) in case of surveillance queries. Different from the approaches above, Miraftabzadeh *et al.* [188] present a privacy-aware framework for identifying and tracking people across surveillance cameras. Each surveillance camera is equipped with computing resources on which face detection is run, and feature vectors or *embeddings* of the faces are generated. The cameras send the embedding vectors to edge servers, which aggregates the vectors from different cameras within its range. The actual recognition of faces occurs in the cloud. Privacy is preserved as only the embedding vectors (and not images) are sent to edge servers and the cloud.

H. Live Streaming

Live streaming applications such as Facebook Live,[14] Periscope,[15] and Twitch[16] allow users to stream live video content from their smartphones and other handheld devices.

[14]https://www.facebook.com/facebookmedia/solutions/facebook-live

[15]https://www.pscp.tv/

[16]https://www.twitch.tv/

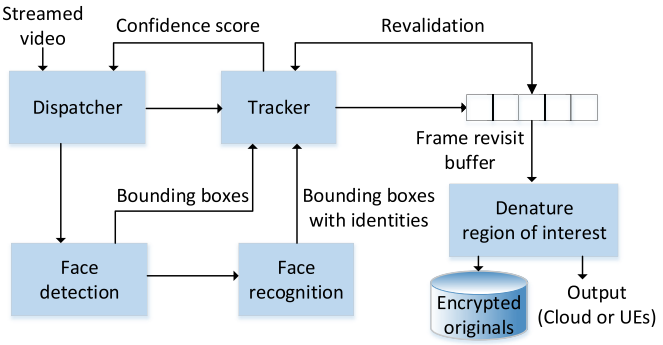

Fig. 18. A denaturing pipeline for blurring or removing privacy-sensitive data from videos [187].

Live streaming systems need to ingest large volumes of video content from the UEs (broadcasters), transcode the content, and adaptively stream the videos to multiple viewers from edge servers close to the viewers. Transcoding is required to provide the appropriate format to viewers based on their device capabilities and quality of their network links. Such systems differ from video-on-demand services as content must be transcoded and delivered to users with a very short end-to-end delay (e.g., 100 ms [189]). Moreover, the broadcasts are spontaneous in nature (i.e., users can stream videos whenever they want); thus, decisions to transcode streams need to be taken in real-time based on the quality of the network link and availability of computing resources for transcoding. An analysis of user traces from existing cloud-based live streaming applications demonstrate the need for edge-C3 to provide localized resources for such systems. For instance, Ma *et al.* [190] find that 45% of the computing resources are consumed by broadcasts which are *all* viewed by users in the same geographic region as the broadcaster [190]. Raman *et al.* [191] demonstrate that close to 40% of broadcasts are not viewed at all; however, current systems still upload these streams to distant cloud data centers resulting in unnecessary use of cloud resources and congestion in the backhaul links.

To address the aforementioned problems, some recent works discuss the use of edge-C3 for live streaming. The following studies discuss the assignment of broadcasters to edge servers with the objective of minimizing latency of streaming.[17] Ma *et al.* [190] study the efficient scheduling of broadcasters to appropriate edge regions to minimize latency while keeping operational costs low (i.e., the cost of running the computational resources for transcoding and delivering services). The authors design a matching algorithm with a classic many-to-one matching to assign broadcasters to edge regions. If necessary, the broadcasters are re-assigned to different regions to balance load while meeting QoS requirements, and until a Nash-stable solution is obtained. The proposed solution is evaluated by using traces from a live streaming platform. The authors find that an edge-based solution reduces latency by 35% as compared to a cloud-based system. Chen *et al.* [192] focus on the problem of choosing both the bitrate of the uploaded video and the edge server where the videos are

[17]See Section V for a discussion about the downlink aspects of live streaming.

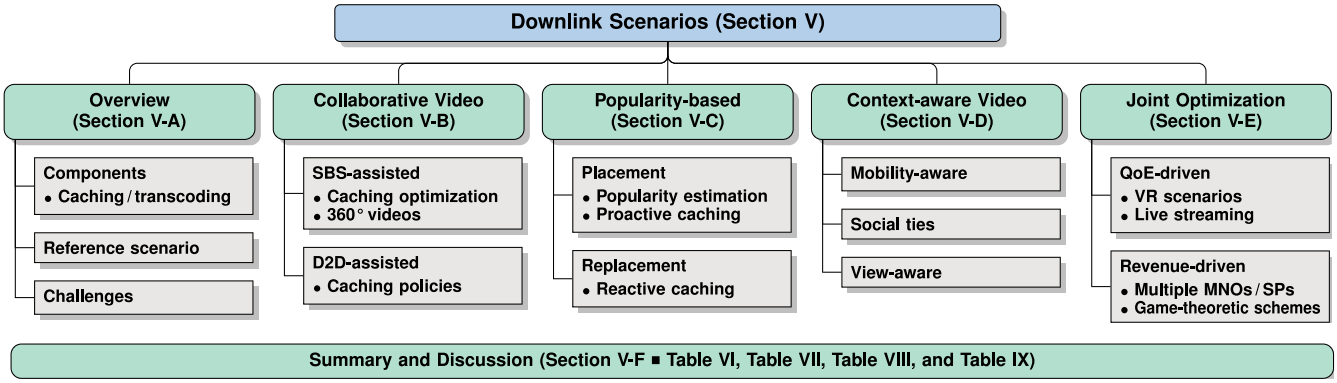

Fig. 19. Organization of the content in Section V.

uploaded and transcoded. The authors propose an optimization problem to choose the bitrate and server while minimizing the end-to-end latency and maximizing the bitrate of all viewers. Their system model also includes routing of videos between different servers and the choice of edge servers for viewers. They present polynomial time heuristic algorithms to solve the problem and evaluate the performance through trace-driven evaluation. Their solution improves latency and bitrates of viewers as compared to baseline approaches that simply choose the nearest edge server.

Some studies propose the use of smartphones for transcoding and distributing live streams. To this end, Zhu *et al.* [193] focus on the problem of choosing UEs for transcoding as well as incentivizing them. In particular, the authors propose a greedy algorithm to select UEs and payment schemes to offer such transcoding services. The objective is to lower the costs of delivering such applications and reduce the end-to-end latency. In their system, a local edge server is responsible for assigning tasks to the UEs, ingesting and forwarding source videos, and recollecting transcoded videos. On the other hand, Dogga *et al.* [189] focus on the operational aspects of live streaming systems that also allow users to distribute videos in a peer-to-peer manner. In particular, they describe a multicast tree-based system wherein users demanding a particular bitrate are modeled as a distributed balanced tree. The leader/root of each tree transcodes video frames and distributes the video to its children in the tree.

All the articles reviewed above use a trace-driven approach to evaluate their solutions, typically using a dataset from Twitch, whereas a more comprehensive dataset (including QoS metrics such as buffering events) is used in [190].

I. Summary and Discussion

Table V summarizes the key features of the articles surveyed in this section. In particular, it describes whether the computation is carried out on UEs (on-device), edge servers or the cloud, and the models or algorithms used to implement tasks in the analytics pipeline (where available). First, we observe that many articles consider running tasks from the analytics pipeline locally on UEs. In the case of analytics applications, these tasks typically comprise lightweight or specialized models for computer vision or computationally-inexpensive tasks such as tracking objects. In the case of live streaming, these tasks typically comprise transcoding. Next, we observe that most surveyed articles use state-of-the-art CNNs for implementing tasks in the analytics pipeline. The CNNs are modified to enable these models to run seamlessly on resource-constrained devices. Moreover, the models may also be specialized for a particular task (e.g., to detect objects of a specific color or type) to further reduce their resource requirements.

In addition to the computational requirements, the design of edge-assisted analytics systems requires careful consideration of latency thresholds and network bandwidth constraints. Different approaches have been proposed to reduce the uplink bandwidth utilization. First, tasks such as object recognition need not be run on all video frames as there is usually some spatio-temporal similarity between frames. Thus, video frames can be sampled at the application layer or by using hardware-based solutions (e.g., [194]) to run analytics only on a subset of frames. The network bandwidth requirements of offloading can be further reduced by compressing videos or reducing the frame resolution. However, such approaches result in reduced accuracy. This can be balanced by defining application-specific QoS requirements and designing offloading frameworks that balance the trade-off between accuracy and latency, bandwidth, and computing constraints. Many articles use an empirical approach, wherein the tasks are first profiled in an offline phase and then used to optimize task placement. However, such an approach may exhibit local patterns and the empirical estimates may need to be updated over time. This aspect has not been addressed in the surveyed articles. Second, the bandwidth requirements can be reduced by running the computer vision tasks at the UEs and only aggregating the results at the edge or the cloud. However, this is usually limited to only certain types of specialized tasks. Finally, customized video streaming protocols for uplink video streams have also been recently proposed to control frame settings while ensuring a minimum accuracy for inference [67]. Such custom protocols are key to improving the overall performance of systems for video analytics [141].

V. Video edge-C3 in Downlink Scenarios

This section reviews and categorizes recent works on video delivery through edge-C3 in downlink scenarios (Fig. 19).

TABLE V
SUMMARY OF WORKS ON VIDEO ANALYTICS IN EDGE-C3

Class	Ref.	Contribution(s)	Algorithms for analytics	Local	Edge	Cloud
Surveillance	[163]	Optimization problem for offloading and scheduling neural network layers to edge servers	AlexNet, CNNs	×	✓	✓
	[143]	Decision framework for energy-efficient offloading of analytics tasks; energy-aware edge routing algorithm	CNN (ResNet)	×	✓	✓
	[145]	Network architecture for cognitive radio and edge-assisted analytics; spectrum-aware placement of services at edge servers	-	×	✓	✓
	[171]	Framework for video analytics on cameras including controller that dynamically re-configures QoS parameters	HOG, Haar cascade	✓	✓	×
	[165]	Query optimizer that determines optimal placement and configuration of parameters for analytics tasks	CNNs	✓	✓	✓
	[166]	Platform for collaborative analytics; optimization problem for offloading tasks that minimizes response time	OpenALPR	×	✓	✓
	[164]	Query optimizer that determines optimal placement and configuration of parameters for tasks in analytics pipeline	OpenALPR, CNN	×	×	✓
	[167]	Platform for cooperative processing of videos on smartphones	AlexNet	✓	×	✓
	[169]	Cooperative video processing on smartphones; optimal forming of UE clusters and dispatching video chunks to the clusters	-	×	✓	×
	[170]	Video surveillance system at the edge; selecting frames such that number of frames with objects of interest is maximized	Haar cascade classifier	×	✓	✓
Augmented Reality	[172]	Optimization problem to minimize the energy required for offloading tasks from analytics pipeline to edge servers	-	✓	✓	×
	[147]	System for object detection that can run at high frame rates	R-CNNs	✓	✓	×
	[173]	System for continuous vision on smartphones; analytics pipeline for faster processing	CNNs, Viola-Jones detector [153]	✓	✓	×
	[175]	Multi-objective optimization problem for task assignment and frame resolution selection	YOLO	×	✓	✓
	[174]	Optimization problem to maximize the accuracy of tasks considering trade-offs against resource usage	CNNs	✓	✓	✓
	[144]	Task scheduling and offloading to optimize trade-offs between resource usage, bandwidth, and latency	CNNs	✓	✓	✓
	[155]	System for caching object recognition models at edge servers to minimize latency for object recognition	ORB	×	✓	✓
	[162]	System for pre-fetching and caching object recognition models at edge servers to minimize latency	SIFT, SURF and ORB	✓	✓	✓
	[146]	System for low-latency object recognition and tracking integrated with current AR software development frameworks	SIFT and Locality Sensitive Hashing	✓	✓	✓
Drone analytics	[176]	Lightweight and computationally inexpensive CNN for detecting vehicles from drone videos	ShuffleDet (new CNN model)	✓	×	✓
	[177]	Evaluation of CNN models on-board drones	YOLO and TinyYOLO	✓	×	✓
	[179]	Video processing from drones that saves wireless bandwidth	CNNs	✓	✓	×
	[181]	Automatic drone coordination for streaming live sports events	-	×	✓	×
	[180]	Algorithm that offloads frames from drones based on its predicted path to maximize application utility	SIFT	✓	✓	×
Vehicular	[183]	Vehicular analytics platform on cars and edge servers	CNNs	✓	✓	✓
	[186]	Estimating availability of parking spots by analyzing videos collected from smartphones in cars	Haar-like features	✓	×	✓
	[185]	System to track paths of vehicles across a network of fixed and mobile cameras	YOLO	✓	×	✓
	[184]	Cooperative distributed analytics from multiple video sources with shared data views and service composition	-	✓	✓	✓
Privacy	[187]	Crowd-sourced video analytics with user-specific privacy policies to edit / blur specific objects	Haar-like features	×	✓	✓
	[188]	Privacy-aware live video analytics that selectively blurs faces based on user-defined policies	Custom DNN-based face detector	×	✓	✓
	[189]	Privacy-aware platform for identifying and tracking humans across a network of surveillance cameras	Pipeline of three CNNs	✓	✓	✓
Live streaming	[191]	Optimal assignment of broadcasters to edge servers to minimize latency and operational costs	-	✓	✓	✓
	[194]	Selection of UEs to perform trancoding of videos; payment schemes to incentivize UEs to transcode videos	-	✓	×	✓
	[190]	System comprising UEs that transcode and distribute videos to followers in a peer-to-peer manner	-	✓	×	✓
	[193]	Optimal selection of edge servers and upload bit rates to minimize latency and maximize quality of live streams	-	✓	✓	✓

First, it describes the implications of video streaming and processing through resources deployed at the edge, in the specific context of wireless networks (Section V-A). Next, it proposes a new taxonomy for video edge-C3 techniques and introduces the state-of-the-art in each category by highlighting the most important contributions. Specifically, we

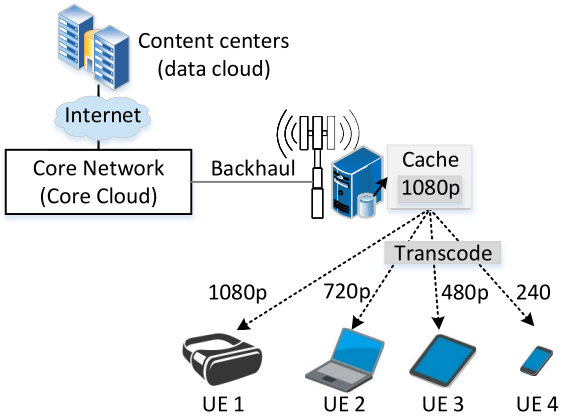

Fig. 20. A general video edge-C3 scenario.

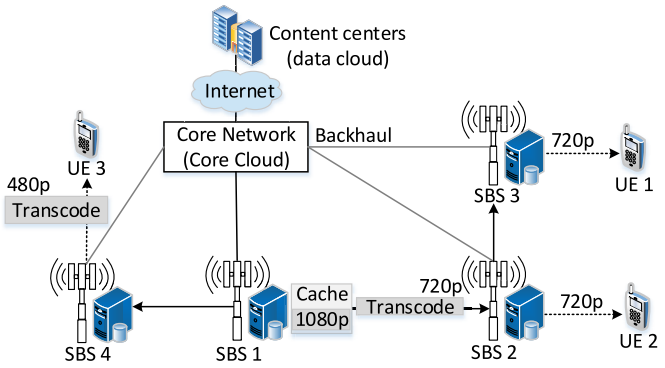

Fig. 21. An SBS-assisted video edge-C3 scenario.

consider *collaborative* approaches, wherein network elements (e.g., SBSs or UE) explicitly cooperate for resource allocation in video delivery (Section V-B). Next, we examine *popularity-based* schemes that (re)allocate storage and computing resources based on how popular videos are among sets of UEs (Section V-C). Moreover, we address how *contextual information* support video delivery based on knowledge of UE mobility, users' social ties, or viewport for 360° videos (Section V-D). Then, we focus on joint optimization for resource allocation based on two primary criteria, namely, *QoE* of users and *revenue* through pricing/trading in a market-based setting (Section V-E). Finally, we conclude by a summary and comparison between the considered approaches (Section V-F).

A. Overview

Edge-enabled video streaming and delivery for downlink scenarios aims at utilizing the edge-C3 resources to provide cost-efficient and seamless video streaming services to client UEs in next-generation wireless networks [195]. The main rationale is to cache popular videos with an appropriate quality (e.g., the most downloaded quality) at edge devices (i.e., base stations or UEs). Once a video requested by a UE is hit at the edge (e.g., in a base station or a neighboring UE), the video segments (or chunks) are transcoded to appropriate bitrates in real-time (e.g., based on the current wireless link quality) and transmitted to the UE. Fig. 20 illustrates a general video edge-C3 scenario for downlink streaming, in which different qualities of a (cached) video are transmitted to UEs with different service requirements through an edge server. In particular, the requested video is transmitted to UE 1 without any transcoding, whereas the video is transcoded to appropriate qualities before it is streamed to UEs 2-4.

Optimal allocation of edge-C3 resources to simultaneous video streaming tasks in real-time is challenging because specific allocation policies result in different performance trade-offs (e.g., QoE versus traffic or latency) [195], [196] and economic models [197]. For instance, by allocating more computing resources, edge servers can transcode and send videos with different qualities to UEs rather than fetching them from the network backhaul, thereby reducing the network backhaul traffic. In addition more videos can be cached at the edge by increasing the storage space in edge devices. As a consequence, the number of video requests served by edge devices increases, thereby reducing the download latency observed by UEs [198].

B. Collaborative Video Edge Delivery

Video delivery through edge-C3 resource involves complex network systems with different elements involved (recall Fig. 20). As a consequence, effective allocation of resources requires coordination between base stations and UEs. In the following, we focus on collaborative approaches that leverage explicit cooperation between network elements. In particular, we distinguish between: *SBS-based* approaches, as performed exclusively by MNOs; and *D2D-assisted* schemes, wherein UEs actively participate in video delivery according to the crowdsourcing paradigm.

1) SBS-Assisted: Fig. 21 illustrates an SBS-assisted video edge streaming scenario in which SBSs cooperate with each other to serve the video requests of their UEs. It is assumed that a 1080p quality video cached in SBS 1 is requested in different qualities by UEs 1-3, where each UE is associated with a distinct SBS. UE 1 requests the video with the same quality; thus the video is transmitted to UE 1 through SBS 3. The video requested by UE 2 is transcoded to quality 720p in SBS 1 and then transmitted to the UE through SBS 2. In a different scenario, the video with 1080p quality at SBS 1 is first transmitted to SBS 4. Next, it is transcoded to 480p quality by SBS 4 and streamed to UE 3. As highlighted in this scenario, the coordination among SBSs in collaborative video streaming is non-trivial, especially when neighboring SBSs have different traffic loads.

Different SBS-assisted video edge delivery mechanisms have been proposed in the literature. Octopus [199] is a hierarchical video caching strategy in C-RAN in which the video requests of UEs are first looked up in their associated SBS and then in their neighboring SBSs. The problem is formulated as a delay-cost optimization, where proactive cache distribution and reactive cache replacement algorithms are proposed to solve the problem. The experiments using real-world YouTube data shows that Octopus improves cache hit ratio, video delivery delay, and backhaul traffic load significantly. Qu *et al.* [200] study how multiple bitrate videos should be cached in SBSs proactively so that the cooperation

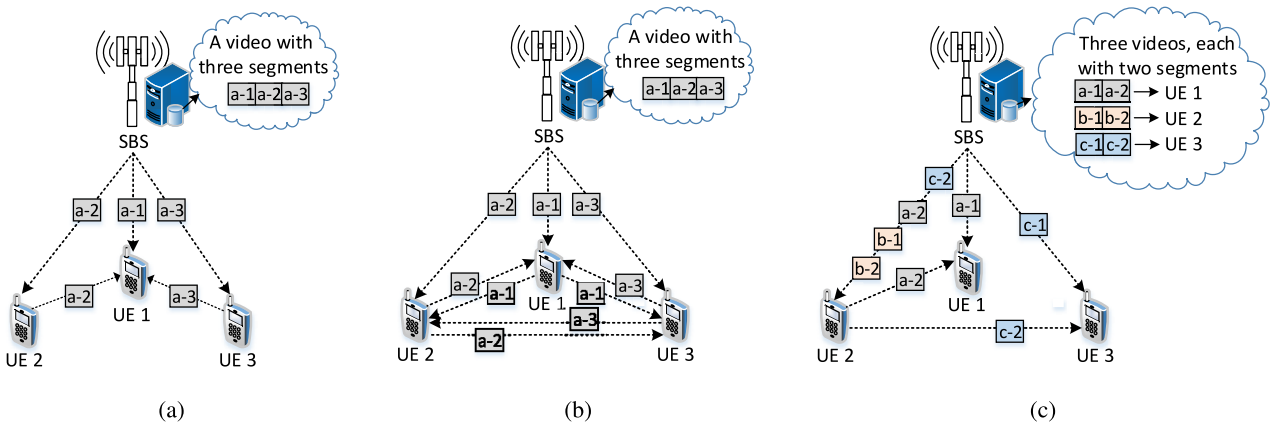

Fig. 22. D2D-assisted video edge-C3 scenarios (adapted from [197]).

between SBSs in video delivery maximizes UEs' QoE function. The QoE function is defined as a UE's perceived QoE and received bitrate. Analytical and experimental results show that the proposed greedy algorithm achieves an approximation ratio arbitrarily close to 1/2, which outperforms existing benchmark solutions (such as FemtoCaching [201]) under non-linear and linear QoE functions. Ao and Psounis [202] design a video delivery architecture by combining the idea of FemtoCaching [201] and SBS cooperation in which clusters of neighboring SBSs are formed dynamically to cooperatively deliver UEs' video requests. Specifically, a cross-layer optimization (i.e., video placement in the application layer and cooperative transmission in the physical layer) is proposed to jointly optimize the video caching and transmission. Liu *et al.* [203] address collaborative video caching and delivery wherein different segments of a video are streamed to UEs by different SBSs. When one UE requests a video, a greedy algorithm selects a proper SBS for downloading the segments of the video, and when multiple UEs request to watch a video, the greedy algorithm is combined with interference alignment method to jointly reduce the video freezes while improving UEs' QoE. Yu *et al.* [65] propose a centralized collaborative caching mechanism in which appropriate video bitrates are selected for streaming with the aim of maximizing the number of video requests served while minimizing the transmission cost. The problem is formulated as a joint video caching and scheduling optimization, for which a two-stage rounding-based algorithm is proposed. Simulation results show that collaborative caching significantly reduces the delivery delay but not the number of served UEs.

Next, collaborative caching and delivery has also been studied in the context of 360° videos. Maniotis *et al.* [204] study a collaborative approach with SVC encoding of 360° videos to decide which tiles and layers of videos are cached in each SBS and which route to deliver them to interested UEs. Decoupling the problem into caching and routing optimizations, Lagrange partial relaxation method is applied to solve the problem. Dai *et al.* [205] propose a synthesis-based VR caching scheme in C-RAN. Synthesis involves combining multiple views (e.g., texture and depth) to generate a multi-view 360° video. The authors propose an architecture where edge servers are deployed in the BBU pool and RRH to synthesize views and serve the 360° video requests of UEs. The problem is formulated as a hierarchical collaborative caching, where an online MaxMinDistance algorithm is applied to find optimal video tiles for caching. The experiments show that the proposed solution maximizes the cache hit ratio and UEs' QoE while minimizing the backhaul traffic.

2) D2D-Assisted: In this model, UEs in close proximity cooperate with each other via short-range RATs (e.g., Bluetooth or Wi-Fi) [34] to serve the video requests of each other. Crowdsourced mobile streaming (CMS) [206] is a common D2D-assisted communication model in which UEs with high-quality Internet access share their resources (e.g., bandwidth) with those in proximity that have slower or unreliable Internet connections [197].

Fig. 22 illustrates different D2D-assisted (or crowdsourced) video streaming scenarios in which UEs 1-3 in proximity download video segment from SBSs and share them among each other cooperatively. In particular, Fig. 22(a) shows a video with three frames delivered to UE 1, where some segments of the video are delivered to UE 1 by UEs 2 and 3. For the same video, Fig. 22(b) illustrates the case where UEs first download video segments and then share them with each other. Finally, Fig. 22(c) shows the delivery of three different videos to distinct UEs; UE 2 has a higher-speed and reliable Internet access, thus, it downloads and delivers some segments of other videos to UEs 1 and 3 (according to the CMS paradigm).

Different techniques have been proposed to realize D2D-assisted video caching and computing at the wireless edge. Some studies have investigated the impact of cache size and video popularity on the performance of D2D-assisted video delivery. Golrezaei *et al.* [207] extended the idea of FemtoCaching [201] to D2D-assisted video delivery in which UEs with caching capabilities play the role of mobile helper nodes and upscale the network capacity with low deployment cost. The experiments demonstrate that D2D-assisted video delivery achieves 1 to 2 orders of magnitude increase in network capacity. However, how to stimulate UEs to participate in video relaying and coordinate their cooperation are the main challenges in D2D-assisted video delivery. Zhou [208] proposes a D2D-assisted video delivery system wherein UEs

make caching decisions by estimating the popularity of videos using information from neighboring UEs. Moreover, UEs can vary their mobility and transmission parameters based on the availability of videos. The proposed scheme outperforms common practical video streaming methods in terms of robustness and efficiency.

Some studies have explored the impact of caching policies of UEs and video size on the performance of video streaming. Kim *et al.* [209] consider a scenario in which each UE caches a subset of video files from a library. Next, UEs in proximity fetch their requested videos through D2D communication from their neighboring UEs. A quality-aware stochastic DASH streaming algorithm is designed for link scheduling and streaming phases. The experiments show a considerable gain in terms of fair link scheduling with respect to off-the-shelf streaming components. The experiments in [210] show that the current fixed-thresholding mechanisms for content caching in Android devices cannot effectively balance the trade-off between the cost of unconsumed content and the QoE. To resolve this issue, an adaptive thresholding solution is proposed to efficiently cache content in UEs. Zhang *et al.* [211] consider the high bit cost of video caching over flash memory as opposed to conventional (magnetic) hard drives. Accordingly, they design a fault-tolerant solution to enable the use of lower-cost (thus, less-reliable) flash memory chips; their solution also reduces the complexity of transcoding by leveraging both video characteristics and the physics of flash memories.

C. Popularity-Based Video Edge Delivery

The popularity of videos viewed by UEs is highly predictable [212], [213]. Hence, the video viewing behavior of UEs can be collected and locally analyzed by edge servers (e.g., at SBSs) to proactively decide resource allocation in edge-C3, particularly of storage at SBSs. Popularity prediction of content in a general context (i.e., other than video) has been extensively studied [214]. In the following, we only consider works specifically targeting video in wireless networks. We classify the surveyed article according to their focus: *placement* of resources in the network, or their *replacement* if previously allocated.

1) Video Placement: Hou *et al.* [215] propose a light-weight transfer learning technique to estimate the popularity of videos through base stations with short training time. The motivation is to transfer the popularity knowledge from previous learning tasks to a target task when the latter has limited high-quality training data. The experiments show that the proposed method improves the cache hit ratio between 17%-117% while reducing average transmission cost by 15% compared to alternative caching solutions. Müller *et al.* [216] propose an online multi-armed bandit algorithm to learn context-specific popularity of videos. The devised solution dynamically updates cache placement by observing contextual information of UEs. In addition, the authors derive a sublinear regret bound which characterizes how fast the proposed solution converges to optimal cache placement. Numerical evaluation using real-world datasets demonstrates that the proposed method increases the cache hit rate by 14% with respect to the state-of-the-art. Chen *et al.* [217] apply echo state networks – a type of recurrent neural network – in a C-RAN setting, which leverages patterns of UEs' content requests at each base station to predict video popularity and UEs' mobility at the BBUs. Experimental results using real-world video traces show that the proposed solution increases the total effective capacity by 27.8% and 30.7% with respect to two random-caching with clustering and random-caching without clustering, respectively.

Once the popularity of videos in each edge server is predicted, proactive caching techniques are applied to identify which videos need to be cached at each base station or edge server. StreamCache [218] leverages the video popularity to provide proactive online caching in ICNs, where the popularity of videos is modeled using a Zipf distribution. StreamCache updates the popularity of videos in rounds using the most recent video request statistics. The objective is to fill the gap between offline theoretical optimal solution and the real-world application. Simulations show that StreamCache obtains an average video throughput per UE that is very close to optimal offline caching. Hoiles *et al.* [219] propose an adaptive video caching algorithm that leverages both the short-term and long-term video popularity to maximize cache hit ratio. In particular, a non-parametric learning algorithm is applied to characterize preferences of YouTube viewers and predict their video request probability in the short term. In addition, a regret-matching algorithm is applied to provide base stations with caching decisions for the long term. Liu *et al.* [220] analyze 10 million video requests of six popular video SPs in China to derive optimal regions for deploying cache-enabled base stations and to determine what content is cached in each location. The authors propose new metrics such as view concentration, popular video number, cache revenue, and popular topics. Their evaluation shows that considering these metrics improves the average cache hit ratio up to 30%. Carlsson and Eager [221] analyze YouTube video data collected over 20 months to design on-demand edge video caching policies. Specifically, a workload model is applied to study the ephemeral popularity of videos, i.e., videos that are watched once or a few times in a particular period. Finally, Hong and Choi [222] propose caching the beginning (called prefix) of popular videos on the UEs themselves. The goal is to minimize the startup delay by building a library of prefixes of videos based on the user's interests. Accordingly, they derive optimal prefix sizes that are to be cached to minimize average delay and storage space.

2) Video Replacement: The popularity of videos in some applications can change frequently, which implies that some already-cached content need to be replaced in order to reflect the newer video demand. Thus, a reactive video replacement policy should be applied periodically to replace some already-cached low-popular videos with newer and more popular items so that the cache hit ratio is maximized. To this end, conventional cache replacement strategies include the least recently used (LRU) and least frequently used (LFU) [223]. The LRU scheme replaces the least recently used content with newer items, whereas the LFU method replaces the least frequently used content with newer content. However, methods leveraging

LRU/LFU can result in poor performance in wireless edge caching scenarios because UEs may be associated with different base stations at different time periods (e.g., due to their mobility or time-varying features of wireless channels).

To deal with above-mentioned challenges, novel studies have leveraged the RAN information to improve the performance of video replacement at the wireless edge. Mokhtarian and Jacobsen [224] propose a flexible ingress-efficient algorithm to enhance the LRU strategy by forecasting future requests of UEs and considering the varying traffic load at the edge devices. The experiments show that the proposed scheme increases the caching efficiency by up to 12% during peak video traffic periods. Ahlehagh and Dey [225] propose a combined proactive and reactive video-aware resource scheduling technique which utilizes UEs' profile information to maximize the number of parallel video sessions served by base stations while satisfying UEs' QoE and minimizing the stalling. The experimental results show that the proposed scheme improves the network capacity by 50% compared to video replacement methods that use LRU. Qiao *et al.* [226] develop a video replacement method to support highly-mobile users. Their solution leverages UEs' video request statistics to identify videos to be cached at each mmWave base station so that the handoffs of UEs are minimized. Specifically, a Markov decision process is applied to dynamically allocate proper cache memory space of each SBS to its associated UEs. Zhan and Wen [62] study SVC video placement at SBSs using the RAN topology information, in addition to the popularity and structural characteristics of layered videos. A heuristic solution with convex relation is proposed to solve the integer programming problem, where the objective is to minimize the average download time under the constraint of cache size at each SBS. Claeys *et al.* [227] propose cache replacement algorithms for video streaming by using not only the temporal features of videos but also user behavior, i.e., in watching consecutive episodes of video series. Based on trace-driven VoD data, simulation results show that the proposed caching strategies improve the state-of-the-art with a 20% increase in the cache hit rate and 4% lower bandwidth usage.

D. Context-Aware Video Edge Delivery

The video delivery process can significantly benefit from information other than the sheer content itself. Such information is mostly represented by the context of UEs, which include their activity (e.g., spatio-temporal network utilization and mobility pattern) and intrinsic characteristics (e.g., social ties and view within the video). We group recent works on context-aware video accordingly under the following categories: approaches that rely on knowing the *mobility* of UEs; schemes leveraging *social ties* between different users; and solutions specifically considering the portion of video that is interesting for a user, particularly, the *view* or gaze in videos.

1) Mobility-Aware: Streaming videos to highly-mobile UEs is extremely challenging because different segments of a video viewed by a UE might be fetched from different base stations as the UE passes through their coverage areas [223]. Furthermore, frequent quality switching may occur in adaptive streaming due to the time-varying quality of wireless links, which in turn negatively impact users' QoE. A common solution is to leverage movement information about the UEs to predict their future mobility (e.g., moving speed and direction) [213].

In this context, the majority of mobility-aware video edge caching and streaming studies have addressed vehicular network scenarios. Zhang *et al.* [228] propose a mobility-aware hierarchical caching architecture in which smart vehicles store popular video content by explicit cooperation with SBSs. Moving vehicles communicate with each other or with the roadside communication infrastructure to facilitate efficient delivery of content to mobile UEs. Experimental evaluations show that the proposed solution improves the performance of content delivery in term of delivery latency. Guo *et al.* [229] propose a video caching and streaming solution in vehicular networks that relies on two time-scales. Specifically, video quality adaptation and cache replacement are performed at a larger time-scale, whereas the transmission of video segments is carried out at a small time-scale. The objective is to maximize the weighted sum of video quality delivered to UEs while reducing the backhaul traffic. Dai *et al.* [230] analyze video caching in a C-RAN in which the centralized BBU pool leverages the UEs' mobility and video popularity information to predict the next cell visited by each UE, so as to efficiently allocate caching and computing resources to base stations. Experimental results demonstrate that the proposed solution improves on traditional caching solutions by 20% and 16% in terms of average transmission delay and cache hit rate, respectively. Kumar *et al.* [231] propose a QoS-aware hierarchical Web caching scheme for video streaming in vehicular ad hoc networks. Their solution takes into account two metrics, namely, load utilization ratio and connectivity ratio. Simulation results show that the proposed scheme reduces communication costs by about 16% and increases the cache hit rate by nearly 9% with respect to conventional approaches. Vigneri *et al.* [232] propose to use vehicles as mobile relays for low-cost video delivery without imposing any streaming delay on UEs. Simulations using real traces – for both video popularity and vehicular mobility – determine that up to 60% of traffic load on the cellular network is reduced by caching content in the vehicular infrastructure.

2) Social Ties: UEs with strong social ties or similar interests exhibit similar mobility and content demand behaviors [213], [233]. Hence, the social features of UEs can be leveraged to predict their preferences and future interactions in video edge delivery. Su *et al.* [234] propose a social-aware caching algorithm for SVC videos in which multiple groups of users with social ties compete with each other for the number of layers they request to cache. Specifically, a non-cooperative game is designed to model the competition among user groups with the aim of maximizing their total profit. Social-Forecast [213] leverages the propagation patterns of content on social media to predict the popularity of videos for different UEs. The objective is to maximize the forecast reward by jointly optimizing the accuracy of predictions and its timeliness. The analytical and simulations-based results reveal that Social-Forecast improves the prediction reward by more than 30% against approaches that use no context information.

Wu *et al.* [236] explore mobility patterns and social aspects of UEs to design a pricing-based system for video edge caching and delivery. In particular, they elect so-called core users to collaborate with an SBS and distribute videos to other UEs through D2D communications. Zhao *et al.* [237] leverage the history of UEs' requests and their social similarity to optimize cache hit ratio and transmission delay in D2D-assisted video edge caching and streaming. The cache replacement problem is formulated as 0-1 knapsack problem which is solve using Lagrangian multipliers to maximize the cache hit rate while minimizing the transmission delay. Sermpezis *et al.* [238] introduce the concept of *soft cache hits* based on which UEs get recommendations on similar videos rather than the requested one, when the latter is not cached at SBSs. The authors argue that UEs are likely to accept the recommended alternative since the majority of video content in the Internet is entertainment-oriented.

3) View-Aware: The current view of a user can be used to improve its QoE in scenarios such as cloud gaming and streaming 360° videos. This is known as *foveated video streaming*, wherein the downlink bandwidth requirements are reduced by streaming high quality video at the viewer's gaze location in a frame and low quality video elsewhere. This relies on the fact that the acuity of the human visual system is highest in the gaze direction and decreases exponentially away from the gaze [239]. Thus, foveated video streaming can be imperceptible with suitable parameterization. We first discuss some approaches which rely on foveated streaming for conventional videos and then discuss them in the context of 360° videos. An additional eye tracker is required for traditional videos, whereas newer VR head-mounted displays contain built-in eye trackers. Thus, a view-aware approach to caching and streaming is a promising method to support streaming of 360° videos.

Ryoo *et al.* [240] design a foveated video streaming solution using a Web-camera based eye tracker and a tile-based encoder. The proposed solution divides a video frame into multiple spatial tiles and encodes each tile in multiple resolutions. The resolution of a tile delivered to the streaming client is proportional to its spatial proximity to the gaze location reported by the client. Illahi *et al.* [241] design a foveated video streaming solution for cloud gaming, wherein a consumer-grade eye tracker at the gaming client is used to report the players' gaze to a cloud gaming server deployed in the edge-C3. The cloud gaming server is configured to encode the gameplay video with a quality dependent on the gaze location. Such a solution reduces the downlink bandwidth requirement by upto 50% with minimal impact on players' quality of experience.

In the context of 360° videos, properties of the field-of-view or viewport of the user are used to cache or proactively send high quality frames, such that the users' QoE is improved. Maniotis *et al.* [204] consider an optimal caching scheme that uses layered and tile-based encoding of 360° videos. Specifically, each tile is encoded into layers of different qualities. The tiles belonging to popular viewports are cached with higher quality in the edge-C3, whereas the remaining tiles are cached with a lower quality. The authors propose an algorithm to determine the optimal set of tiles and their qualities to cache in the edge-C3 considering the limited storage space. Mahzari *et al.* [242] propose a tile-based caching policy at the edge servers, that additionally determines which tiles to replace from the cache when capacity is exceeded. In their system model, the UE chooses the quality of the requested tiles based on network conditions and informs the edge-C3 whether the requested tiles are within its viewport or not. These parameters are used by the edge server to learn a probabilistic model of the tile and quality requests. Such a model is used to make caching decisions, i.e., which tiles and qualities are to be cached or replaced. Their proposed solution outperforms the cache hit ratio as compared to LRU and LFU by 17% and 40% respectively. Papaioannou and Koutsopoulos [243] consider an optimal caching scheme for tile-based 360° video streaming. The authors examine both layered and non-layered video encoding scenarios where each tile has multiple possible resolution levels and each level has different request frequencies based on historical viewing data. The authors propose a solution to maximize the caching of tiles at the resolution level with the highest request frequency, while considering foveated display of the tiles. Different from the above approaches, Perfecto *et al.* [244] propose a proactive scheduling algorithm of 360° video frames to UEs based on predicted viewports. Specifically, they consider a scenario where high-quality (e.g., HD) frames are already cached at the base stations, and SD frames on the UEs themselves. The UEs report their viewports and video indices to an edge server. This information is used in the edge-C3 to predict UEs' future viewports as well as to cluster UEs (based on their overlapping viewports and physical locations). The edge server then proactively sends high quality frames to clustered groups from the appropriate base station. Such an approach allows the streaming service to maintain low latency of streaming and prevent VR sickness, while maximizing the quality of streaming. Lungaro *et al.* [245] propose a gaze-aware video streaming solution for 360° videos using a head-mounted display with an eye tracker. The proposed solution utilizes a server for video tile provisioning and streaming that can be deployed in the edge-C3. The authors propose modifications to the HEVC encoding standard to support foveated streaming of 360° videos. They determine through user studies that the downlink bandwidth is reduced by 60% to 80%.

E. Joint Optimization of Video Edge-C3 Resource Allocation

Complex resource allocation problems arise in video streaming systems, due to trade-offs between the utilization of different resources (e.g., network bandwidth, caching, and compute) and the QoE of UEs. For instance, videos may be streamed with a higher quality at the expense of increased network bandwidth. On the other hand, the choice of video quality levels depend on both the storage capacity and compute capacity (for transcoding in case the requested quality is not cached) at the edge server. Finally, an increasing emphasis is placed on lowering the energy consumption of UEs in emerging VR applications based on 360° videos. On the other hand, allocation of edge-C3 resources can also be driven by the goal to maximize the revenue of MNOs and video SPs. For

instance, caching and compute resources in the edge-C3 result in increased operating costs for the MNOs. Furthermore, when many MNOs and video SPs are part of the system, competitive market-based allocation problems emerge wherein MNOs sell caching resources to SPs. The objective is to maximize the revenue while meeting a target quality.

In the following, we classify recent solutions for the joint optimization of edge-C3 resource allocation for video streaming according to the criteria above. Specifically, we distinguish solutions wherein the optimization is driven by the *QoE* of UEs from those primarily addressing the *revenue* of different actors in a market-based (or economic) setting.

1) QoE-Driven: The following articles propose optimization problems to maximize the QoE of UEs while minimizing the utilization of edge-C3 resources. Different combinations of edge-C3 resources are considered in the surveyed articles, which we describe next.

Jin *et al.* [246] study the joint optimization of edge caching, computing (i.e., transcoding), and bandwidth resources for on-demand video streaming. They formulate a constrained optimization problem to minimize the total caching, computing, and bandwidth utilization for each user request. They then derive closed-form solutions for the optimal transcoding configuration and allocation of cache space. They evaluate their solution through simulations and find significant resource savings compared to state-of-the-art approaches. Moreover, they investigate the trade-offs between utilizing different types of resources and how they impact practical video streaming solutions in edge-C3. For instance, they report that if the trancoding costs are high, it is better to fetch content directly from the SP's server rather than caching the high quality representations. Liang *et al.* [94] propose an optimization problem to assign an optimal video quality level to each UE while determining an optimal network path. They also incorporate the computing resources required for transcoding the video streams in case the chosen quality level is not cached at the edge server. A dual-decomposition method is applied to obtain the decision variables (video data rate, computing resource, and network path selection) independently while maximizing the user's QoE. In contrast, Xu *et al.* [247] investigate joint cache allocation and bitrate selection in adaptive video streaming and leave the computing costs as future work. The authors use a combination of a Stackelberg game and matching algorithm to identify videos to be cached in each base station. Mehrabi *et al.* [248] investigate QoE-based traffic optimization in collaborative DASH video caching and streaming. They devise a self-tuned bitrate selection algorithm to maximize the QoE while minimizing both the backhaul and fronthaul traffic. The same authors in [80] jointly optimize the QoE of UEs and the balancing of load between edge servers connected to base stations. They aim to fairly allocate edge computing resources for adaptive video streaming to base stations while maximizing the QoE of UEs. The problem is shown to be NP-hard, thus an auto-tuned parameterization technique is proposed to find a near-optimal solution.

In the context of VR, new types of content (360° degree and 3D videos) have to be streamed to UEs. This results in novel considerations of the caching and computing capabilities of both the edge server and the UEs (head-mounted displays) themselves. Liu *et al.* [249] aim to maximize the quality of the tiles in a viewport for 360° video streaming while minimizing the energy consumption of the UE. First, they provide closed form equations for the transmission latency and energy consumed in different scenarios of 360° streaming. Specifically, they consider different types of network links (both mmWave and sub-GHz bands) and whether the viewport is rendered at the edge server or at the UE itself. Next, they propose a multi-objective joint optimization problem to optimize the video chunk quality, link adaptation, and adaptive viewport rendering. As the proposed problem is NP-hard, the problem is solved using a genetic algorithm. Next, Sun *et al.* [250] study the joint allocation of resources for mobile VR that includes both 3D and 2D content. They analyze the different trade-offs between utilizing both computing and caching resources for delivering VR streams that contain 3D content. Specifically, both 2D and 3D content can be cached at the edge, and the compute resources are used to project 3D to 2D content. Caching 3D content lowers the computing requirement as no projections need to be computed before streaming the content to the UE. However, this comes at the expense of increased storage space; specifically, 3D content requires twice more storage space than regular 2D content. The authors investigate different trade-offs taking into the account the caching/compute capabilities of the UE and devise optimal joint caching and computing policies for streaming such content.

Finally, different resource allocation approaches have been proposed in live streaming scenarios, which have stricter latency requirements. Ge *et al.* [251] propose a cache-based mechanism at the edge for live streaming 4K video that reduces the latency, buffering, and startup delays at the viewer's device. To this end, they propose an edge-based transient holding of live segment scheme that holds back an optimal number x of video segments from the receiver in order to ensure a certain QoE. The edge server than opens up parallel connections to the live source and downloads the segments before the viewers request them. When the local content at the edge server is at least x segments ahead of the viewer's request, the parallel connections are no longer maintained and only one segment at a time is downloaded from the live source to remain ahead of the viewer's request. They evaluate their solution through real-world experiments and show that such an approach eliminates buffering and significantly reduces the live stream latency. Zhang *et al.* [252] aim to maximize the quality of the live stream (in terms of PSNR) while minimizing the latency of the video stream. Their system model takes into account the computing resources required for trancoding and allocation of wireless spectrum to the viewers. They model the problem as a Markov Decision Process; the authors then propose an enhanced version of reinforcement learning to solve the problem. Their solution outperforms baseline reinforcement learning approaches. Finally, Hung *et al.* [253] focus on the assignment of caching space to live streamers to improve the QoE of UEs. They use an auction-based mechanism to optimally assign caching space to streamers taking into account both storage space and backhaul capacity. They

provide low-complexity and scalable algorithms to solve the assignment problem in real-time at the edge server.

2) Revenue-Driven: Video caching at base stations result in additional operational costs to MNOs, particularly in terms of the costs of edge resources (e.g., storage and processing). Thus, resource allocation problems for video streaming can also be studied in terms of the costs to MNOs. Ghoreishi *et al.* [254] formulate the trade-off between the storage cost and bandwidth savings in hierarchical video caching systems as a binary-integer programming model. The objective is to find the optimal cache size at different layers of a hierarchical caching system so that the ratio between the transmission costs and storage cost is minimized. The evaluations show that benefit-cost and cost-efficiency ratios are improved more than 43% and 38%, respectively. Poularakis *et al.* [37] address the joint optimization of the storage costs and perceived latency for the delivery of SVC videos in HetNets. The proposed framework takes into account different system constraints, such as the backhaul link capacity, the cache size, and wireless capacity of SBSs. Moreover, the framework includes a penalty cost to account for future revenue losses when the UE requests cannot be met due to limited resources. The experimental results reveal that a 10% improvement in video delivery latency may cause about 10% to 30% increase in the operational costs, depending on the network load. Zhou *et al.* [255] study the joint optimization of video caching, transcoding, and communication resources in a virtualized HetNet. In particular, the costs of computing and caching are inversely related. Specifically, when more video versions are cached, the requirement for transcoding (and thus computing) is lowered. Their proposed system uses multicast to simultaneously transmit the same video content to multiple UEs over the same frequency band. They then evaluate the impact of storage and computing capacity on the MNO's revenue. For instance, when the size of the cached videos increase, fewer versions can be cached resulting in lower caching revenue and higher computing cost.

Data sponsoring has been considered as a promising mechanism to increase the number of video streaming subscribers of SPs (and thereby their revenue). Through this approach, video SPs subsidize the UE's cost for watching videos thereby increasing the number of users (and thus, advertising revenue by placing in-video advertisements in exchange for the reduced data access cost). In such a context, Sun *et al.* [256] propose a two-stage decision making process to maximize the revenue of a single SP within a fixed budget that has to be spent on both sponsoring and storage costs. Accordingly, SP determines the edge caching policy in the first stage and the real-time sponsoring decision in the second stage. Simulation results demonstrate that such a joint optimization improves the revenue of the SP by 124%–154%, compared to data sponsoring without edge caching.

The articles described above have focused on single MNOs and single video SPs. In practice, the network infrastructure and edge resources are provided by one or multiple MNOs, which are rented by different video SPs. Due to the limited edge resources, sellers compete with each other over renting a portion of them to deliver quality services to UEs. This implies the creation of a market, where the price of edge-C3 resources is defined based on profit analysis (i.e., based on related costs and revenues) for both resource sellers (e.g., MNOs) and buyers (e.g., video SPs). Generally, the sellers and buyers of edge resources have incomplete information about each other and the network status, thus they have to estimate their expected profit (i.e., the utility and cost) from trading these resources.

Different economic models (e.g., game-theoretic approaches) have been employed to analyze the pricing and trading of edge-C3 resources in wireless networks. The authors in [257], [258] apply a Stackelberg game to model the trading of caching resources between one SP who aims at renting and caching its popular videos in SBSs provided by multiple MNOs. The problem is formulated in terms of social welfare maximization (i.e., the total profit of the video SP and MNOs). Next, the Stackelberg equilibrium is applied to find optimal cache prices while maximizing social welfare. Numerical results reveal that effective resource pricing can maximize the profit of the SP and MNOs. Li *et al.* [259] study a different scenario wherein an MNO leases its edge resources at SBSs to multiple video SPs. The authors also use a Stackelberg game to maximize the social welfare of the system. Analytical results based on stochastic geometry show that the proposed solution achieves efficient resource pricing which matches the empirical data. Dai *et al.* [260] study collaborative multimedia streaming in edge-enabled wireless networks in which *selfish* SPs compete with each other to maximize their individual revenue. Given limited edge caching resources, the authors propose a Vickrey-Clarke-Groves auction to maximize the system social welfare while satisfying economic properties such as incentive-compatibility and truthfulness. Jedari and Francesco [261] propose a double auction method called DOCAT for cache trading of SVC videos between an MNO and multiple video SPs in HetNets. They assume that SPs have different popularity, hence, videos of highly popular SPs are requested by their subscribed UEs more frequently. As a consequence, the value of caches at SBSs is higher for more popular SPs, compared to those that are less popular. DOCAT targets efficient and fair trading through an iterative auction; specifically, the cache of SBSs is segmented and then traded in multiple rounds through a many-to-one matching algorithm. Numerical results based on a real video dataset show that DOCAT maximizes the system welfare while guaranteeing the economic properties of rationality, balanced budget, and truthfulness.

F. Summary and Discussion

Tables VI, VII, VIII and IX summarize the major contributions and key features of the articles surveyed in this section. The tables show that the majority of recent works focused on caching, whereas edge computing for downlink video scenarios is more relevant for emerging use cases such as live streaming and 360° video streaming. This is because videos are typically encoded offline in multiple resolutions for VoD scenarios and thus, do not require further processing (computations). On the other hand, edge computing is important in the context of live streaming, wherein transcoding of live streams may be required to support heterogeneous devices and network links (e.g., transcode to lower quality

TABLE VI
SUMMARY OF WORKS ON COLLABORATIVE VIDEO EDGE-C3

Class	Ref.	Contribution(s)	Objective(s)	Caching	Computing	Collaboration	Mobility	SVC
SBS-assisted	[201]	Cooperative and proactive caching for multiple bitrate videos	Maximizing the linear and non-linear QoE functions (UEs' perceived QoE and received bitrate)	✓	×	✓	✓	×
	[200]	Collaborative hierarchical video caching by exploiting C-RAN functionalities	Improving cache hit rate, UE's access delay, and backhaul traffic	✓	×	✓	×	×
	[204]	Cooperative video caching and transmission in SBSs without incurring high traffic	Improving the transmission delay and reducing the UEs interference	✓	×	✓	×	×
	[203]	Joint cross layer optimization of video caching and cooperative transmission	Near-optimal caching for maximizing system throughput or minimizing delay	✓	×	✓	×	×
	[66]	Collaborative SVC video edge caching in software-defined RAN	Maximizing QoE and minimizing the video transmission costs	✓	×	✓	×	✓
	[205]	Collaborative caching and routing of 360° videos with SVC encoding	Maximizing cache hit ratio while minimizing delivery latency	✓	×	✓	✓	✓
	[206]	A synthesis-based hierarchical collaborative 360° VR caching scheme in C-RAN	Maximizing the cache hit ratio and UE QoE while minimizing the backhaul traffic	✓	✓	✓	×	×
D2D-assisted	[208]	Extending the idea of FemtoCaching by using UEs as mobile helper nodes	Improving video throughput and network capacity without deploying additional infrastructures	✓	×	×	✓	×
	[210]	Caching algorithms for D2D communication in adaptive streaming	Improving video delivery throughput in dense HetNets	✓	×	×	✓	×
	[209]	Distributed D2D video delivery scheme with respect to file size	Demonstrating the efficiency and robustness of D2D video distribution	✓	×	×	✓	×
	[211]	Smart device cache management algorithm using adaptive thresholding	Reducing unconsumed contents and video freezing under low-bandwidth conditions	✓	×	×	×	×
	[212]	Solutions to reduce the cost of video caching in device flash memory	Reducing transcoding complexity by exploiting video and flash memory physics	✓	×	×	×	×

TABLE VII
SUMMARY OF POPULARITY-BASED VIDEO EDGE STREAMING AND DELIVERY APPROACHES

Class	Ref.	Contribution(s)	Objective(s)	Caching	Computing	Collaboration	Mobility	SVC
Video Placement	[216]	A light-weight transfer learning technique to estimate the popularity of videos	Maximizing cache hit ratio while reducing transmission cost	✓	×	×	×	×
	[217]	An online multi-armed bandit algorithm to learn context-specific popularity of videos	Increasing the cache hit rate	✓	×	×	✓	×
	[218]	A recurrent neural network which leverages UEs' content request pattern to predict video popularity and UEs' mobility	Increasing the effective network capacity and users' QoS	✓	×	✓	×	×
	[219]	A dynamic video popularity calculation method using most recent video statistics	Maximizing average online video throughput per UE close to optimal offline caching	✓	×	✓	×	×
	[220]	A video caching algorithm that leverages both short-term and long-term popularity	Maximizing the cache hit rate	✓	×	✓	×	×
	[221]	Definition of different video-specific and popularity-based similarity metrics	Maximizing the overall cache hit ratio	✓	×	✓	×	×
	[223]	A caching strategy to store prefixes of popular videos on UEs	Minimizing the average playback delay	✓	×	×	×	×
Video Replacement	[225]	A flexible ingress-efficient algorithm to enhance the LRU strategy	Increasing the caching efficiency during peak video traffic periods	✓	×	×	✓	×
	[226]	A combined proactive and reactive video-aware resource scheduling technique	Maximize the number of parallel video sessions and UEs' QoE, while minimizing stalling	✓	×	×	✓	×
	[227]	A Markov model to allocate proper cache memory space of each SBS to its UEs	Minimizing the handoffs of UEs	✓	×	×	✓	×
	[62]	A heuristic to study SVC video placement at SBSs using the RAN topology information	Minimizing the average download time under the constraint of cache size at each SBS	✓	×	×	×	✓
	[228]	A replacement algorithm for consecutive episodes of video series	Improving the cache hit ratio with lower bandwidth usage	✓	×	✓	×	×

for UEs with low-bandwidth wireless links). In this context, UEs may also cooperate to share their computing resources to transcode and stream live videos with low latency to neighboring devices. Moreover, edge computing is required for streaming VR content (e.g., to compute projections from spherical to equirectangular coordinates). For instance, edge

TABLE VIII
SUMMARY OF CONTEXT-AWARE VIDEO STREAMING AND DELIVERY APPROACHES AT THE WIRELESS EDGE

Class	Ref.	Contribution(s)	Objective(s)	Caching	Computing	Collaboration	Mobility	SVC
Mobility-aware	[229]	A hierarchical cooperative strategy to cache vehicular UEs' popular videos in SBSs	Minimizing access latency and improving resource utilization	✓	✓	✓	✓	×
	[232]	A QoS-aware hierarchical video caching in vehicular networks	Reducing communication and relay costs while improving cache hit rate	✓	×	×	✓	×
	[230]	A video caching and streaming solution in vehicular networks based on two time-scales	Maximizing the averaged weighted sum of video quality while reducing the backhaul traffic	✓	×	×	✓	×
	[233]	Mobile vehicle video caching for low-cost video streaming services	Minimizing traffic load on cellular infrastructure without any streaming delay	✓	×	×	✓	×
	[231]	A low-cost video streaming technique by using UEs' mobility and video popularity	Reducing the backhaul traffic by 60%	✓	×	✓	✓	×
Social-based	[236]	An SVC video edge caching scheme considering the social interactions of UEs	Maximizing UE utility while improving cache hit rate and video delivery latency	✓	×	✓	×	✓
	[237]	A social-based video popularity prediction method	Jointly optimizing the video popularity accuracy and its timeliness	✓	×	✓	×	✓
	[238]	A social-based cache pricing mechanism for video edge delivery	Improving the effectiveness and reliability of video transmission	✓	×	×	✓	×
	[239]	A video distribution system based on social characteristics of UEs	Alleviating the traffic load in SBSs while achieving reliable video delivery	✓	×	×	✓	×
	[240]	Soft cache hits to recommend similar videos rather than the requested one	Reducing the mobile data traffic while maximizing the cache hit rate	✓	×	✓	×	×
View-aware	[242]	A foveated video streaming system using commodity hardware	Reducing downlink bandwidth usage	✓	×	×	×	×
	[243]	A foveated video streaming system for cloud gaming	Imperceptibly reducing downlink bandwidth requirement	×	✓	×	×	×
	[247]	End-to-end foveated video streaming for VR	Reducing downlink bandwidth	✓	✓	×	×	×
	[205]	Caching of viewports with different qualities for 360° video streaming	Minimizing the cumulative distortion experienced by UEs	✓	×	✓	✓	✓
	[244]	Viewport-aware caching policy for 360° videos	Maximizing the cache hit ratio	✓	×	✓	✓	×
	[245]	Viewport and perceptually-aware caching for 360° videos	Maximizing the cache hit ratio	✓	×	×	×	✓
	[246]	Proactive and viewport-aware streaming of 360° videos	Minimizing delay of streaming	✓	✓	✓	✓	×

computing resources can be used to pre-render complex 3D content and stream such content to resource-constrained VR devices.

Next, popularity-based and context-based video caching has received significant attention from the research community (see Tables VII, VIII). However, none have considered the use of edge computing resources to learn patterns of user requests and determine which videos are to be cached or replaced. Finally, the joint optimization (Table IX) of edge-C3 resources may be QoE or revenue-driven. However, most of economic models have considered simple trading models in video edge-C3 and did not study how the structure of videos (e.g., their encoding models) can affect the cost and utility of SPs and MNOs in video service delivery.

VI. OPEN ISSUES AND FUTURE RESEARCH DIRECTIONS

This section introduces several important open questions and future research directions for video edge-C3 in next-generation wireless networks.

Learning-based video edge-C3: Artificial intelligence (AI), specifically (deep) learning techniques, is expected to play a vital role in delivering low-latency and ultra-reliable video services in wireless cellular networks [262]. For instance, deep learning models can be used to predict the popularity of videos at the edge by utilizing the context and request patterns of UEs connected to the local SBS. Such predictions enable intelligent video placement decisions based on the context of users, which can improve the cache hit ratio and video delivery latency. This is especially beneficial in scenarios where local popularity trends do not reflect the global trends. Thus, predicting content popularity trends at the network edge allows SPs to proactively react to local changes (e.g., to allocate more resources to hotspots). Moreover, training prediction models at the edge removes the need to send private information about UEs to the cloud. In this context, *federated learning* [263] has emerged as a promising solution to enable collaborative model training at the edge servers. Federated learning is a distributed learning approach wherein a global model is learned with updates from multiple distributed devices. Each device (edge server, in this case) updates a model (that can be shared with other edge servers in the region) with training data observed locally. Thus, a popularity prediction model can be created based on contextual information gathered at the edge servers. However, there are several practical open

TABLE IX
SUMMARY OF THE WORKS ON JOINT OPTIMIZATION OF VIDEO EDGE-C3 RESOURCE ALLOCATION

Class	Ref.	Contribution(s)	Objective(s)	Caching	Computing	Collaboration	Mobility	SVC
QoE-driven	[248]	Edge-C3 in media cloud for on-demand adaptive video streaming	Optimizing trade-off between the storage, transcoding and bandwidth costs at the edge	✓	✓	×	×	×
	[95]	Joint optimization of SDN, caching and compute resources for streaming	Maximizing the video experience metric U-video mean opinion score	✓	✓	×	✓	×
	[249]	Joint adaptive video caching and streaming at network edge using Stackelberg game	Improving the cache hit rate and video delivery throughput	✓	×	✓	✓	×
	[81]	Network-assisted adaptive video streaming using MEC facilities	Joint optimization of QoE, fairness, and balancing the utilization of RBs among BSs	✓	×	×	✓	×
	[251]	Panoramic VR video caching and computing in millimeter wave cellular networks	Optimizing the video chunk quality, link adaptation, and adaptive viewport rendering	✓	✓	×	×	×
	[252]	Joint caching and computing of mobile VR over wireless edge networks	Optimizing joint policy to minimize average transmission rate	✓	✓	×	×	×
	[253]	Live streaming 4K videos at the edge through transient holding of segments	Optimizing number of held segments to minimize live stream latency	✓	✓	×	×	×
	[254]	User scheduling, compute and wireless spectrum allocation for live streaming	Improving UE's QoS and minimizing latency	✓	✓	×	×	×
	[255]	Allocation of caching resources to live streamers	Maximizing quality of live streams	✓	✓	×	×	✓
Revenue-driven	[256]	Cache provisioning problem in hierarchical in-network caching	Optimal cache size at different layers to minimize the cost ratio	✓	×	×	×	✓
	[37]	Optimal joint routing and caching policies using SVC and non-SVC videos in HetNets	Optimizing the trade-off between delivery costs and user experienced delay	✓	×	×	×	✓
	[257]	Joint video caching, transcoding and multicasting in virtualized HetNet	Jointly optimizing the utility of computing, caching and communication	✓	✓	×	×	×
	[258]	Joint optimization of edge caching and video sponsoring for content providers	Reducing video delivery cost while increasing the revenue of content providers	✓	×	×	✓	×
	[259]	A Stackelberg game for video delivery in commercialized small-cell caching systems	Jointly maximizing the profit of MNOs and SPs	✓	×	×	×	×
	[260]	A Stackelberg game to study cache trading in a network with an MNO and multiple SPs	Increasing the profit of SPs and improving resource allocation	✓	×	×	×	×
	[261]	A commercial video caching system consisting of single SP and multiple MNOs	Jointly maximizing the profit of the SP and MNOs while improving resource utilization	✓	×	×	×	×
	[262]	A Vickrey-Clarke-Groves auction to model cache trading in a network with selfish SPs	Improving the quality of video streaming while maximizing the social welfare	✓	×	✓	✓	×
	[263]	An action-based cache trading mechanism for SPs owning SVC videos	Maximizing the social system welfare while satisfying the economic criteria	✓	×	×	✓	✓

issues for training models at the edge. First, the impact of limited edge computing resources for training models must be analyzed. Second, the communication overhead with federated learning may be quite large as the model parameters need to be shared and aggregated at regular intervals in the edge servers. Recent studies show that the convergence of the trained model depends on the choice of system parameters such as the frequency of updates and aggregation [264]. Thus, a careful study of such parameters is required for edge-based solutions. Third, the design of prediction models must take into account the trade-off between prediction accuracy and algorithm complexity. Applying highly-accurate popularity prediction algorithms improve the caching performance, but it entails higher computational complexity and thus, increased utilization of computing resources in the edge-C3. Finally, it is important to quantify the benefits of using a localized popularity model at the edge (in terms of cache hit ratio or latency) as a trade-off against the increased computation and latency incurred in the training itself. Based on such a trade-off, MNOs can decide whether to use a localized popularity model or a global one to make caching decisions.

Economics of resource allocation in video edge-C3: From an economic perspective, cost-efficient allocation of edge-C3 resources provided by MNOs to multiple video SPs is non-trivial due to several reasons. First, the revenue and cost of different types of edge-C3 resources for MNOs and SPs are different. For instance, the cost of storage resources at the edge (e.g., SBSs) might be lower than processing resources for MNOs, but it can bring higher revenue to SPs. Thus, it is challenging to allocate both dynamically and economically edge-C3 resources to SPs (i.e., their subscribed UEs) such that the social welfare of the system is maximized. Second, since SPs generally have different popularity (e.g., they have a different number of subscribers), the revenue and cost of edge-C3 resources for different SPs might vary. For instance, the revenue of high data-rate bandwidth can be more significant for popular SPs. Therefore, how to allocate edge-C3 resources to SPs with different popularity is a critical and vital decision

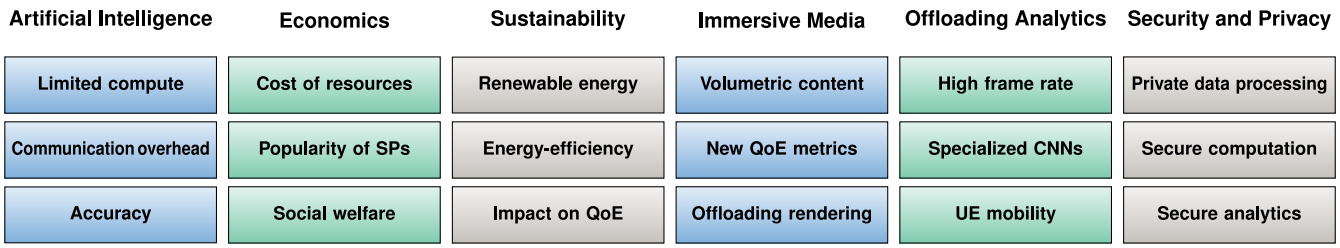

Fig. 23. Main directions for future work by theme.

for MNOs. The problem becomes even more challenging when MNOs and SPs do not have complete information about the profit of each other. Few recent studies have addressed the economics of video edge-C3 resource trading in terms of either caching (e.g., [261]) or computing (e.g., [265]). Nevertheless, how to maximize social welfare in a system with multiple video SPs when they price edge-C3 differently remains an open research problem.

Sustainable video in edge-C3: Infrastructure in edge-C3 systems consume a large amount of energy, which is expected to only increase with the roll-out of dense deployments of edge servers and base stations in future 5G networks. MNOs aim to reduce the energy consumed, both from the perspective of lower operating costs as well as meeting sustainability goals. Thus, new solutions are required to reduce energy consumption as an increasing amount of video content is being consumed and generated by UEs. First, renewable energy sources can be integrated into the edge-C3. Currently, edge servers are mainly powered by energy from brown power grid sources which in turn causes unavoidable environmental concerns in long-term system operation. Renewable sources such as solar or wind help to move towards environmentally-friendly video processing and streaming. In this context, interesting resource allocation problems emerge that require to balance the trade-off between the QoE of streaming, backhaul traffic, and energy consumed [266]. Second, energy consumption can be reduced by switching off under-utilized edge servers. In the context of video edge-C3, switching off servers requires re-directing processing tasks (e.g., transcoding or analytics tasks) from multiple servers to a few edge servers. The design of such a solution requires careful consideration of balancing the trade-off between lowered QoE and reduced energy consumption. Furthermore, determining a switch-off schedule remains an open challenge. For instance, the time intervals can be determined either in an online (e.g., whenever observed traffic is low) or offline manner (based on predicted request patterns). Future research directions include determining the impact of different switching-off schedules on the QoE of video applications and energy consumption. In addition to designing intelligent algorithms, system measurements are required to quantify the trade-off between reducing energy consumption and lowered QoE (e.g., due to processing on edge servers that are further away) for the end users.

Video streaming for emerging applications: AR and VR place new demands on wireless networks in terms of real-time processing of uplink streams with low-latency. We have surveyed state-of-the-art solutions that reduce latency through intelligent application design and caching of data. However, several open research directions still remain. For instance, emerging wireless technologies (such as mmWave in 5G networks) demand new scheduling algorithms to transmit 360° videos to UEs with low latency [267]. Moreover, none of the surveyed articles have considered the end-to-end design of live streaming, wherein the edge server adapts the video streams based on both uplink and downlink bandwidth capacities. Additionally, new forms of video content are being generated today. For instance, *volumetric videos* [268], comprising three-dimensional content in the form of volume pixels or 3D meshes, are increasing in popularity. Such content can be viewed on both smartphones and head-mounted displays, and provide a wider range of interactions compared to traditional or even 360° videos. Specifically, volumetric videos provide users with 6 degrees of freedom, allowing them to change even the orientation (yaw, pitch and roll) of their viewport. The enhanced capabilities of 5G networks are expected to support the streaming of such content over the Internet. This gives way to several new applications, such as immersive telepresence and live streaming of concerts. However, streaming volumetric content is challenging as it is not possible to simply buffer frames at the client device as users may zoom-in or rotate the 3D content when desired. Thus, traditional video streaming solutions (e.g., DASH, WebRTC, HAS) require modifications to support such interactions with a small latency and adequate QoE for end users. To this end, new QoE metrics are also required to evaluate the performance of streaming solutions. Finally, the heterogeneity of viewer devices (smartphones and head-mounted displays) mean that all devices may not be able to decode and render 3D content. Edge-based solutions are ideally suited to provide the computational resources for such applications with very low latency [268]. To this end, new algorithms are required to determine when to render content at the edge server or at the UE based on the available network bandwidth, computational resources and energy available at the UE.

Offloading video analytics tasks: The surveyed articles demonstrate the importance of edge computing to enable real-time analytics on live video streams. There are still several open research challenges in designing efficient edge-assisted systems for analytics. For instance, running analytics tasks such as object recognition at high frame rates is still an open problem. As reviewed in this article, several works propose the use of specialized CNN models at the edge to speed-up the inference. Such specialized models are trained offline based on known application characteristics (e.g., detect object of a

certain color) or user request patterns. However, contextual information and the spatio-temporal locality of requests could be used to automatically specialize CNN models deployed at the edge. For instance, the CNN models can be re-trained at the edge based on recently observed input video streams and user requests. This would increase the efficiency of analytics by reducing the latency of inference for similar future requests at the edge-C3. Second, real-time analytics in the presence of high mobility of users – for instance, in vehicles – is an important open issue. Specifically, offloading decisions require careful consideration of where tasks are offloaded, where the UEs will receive the computational results, and whether applications (or tasks) need to be migrated between edge servers. Designing task offloading frameworks for mobile users and with strict latency constraints required by video analytics has yet to be fully addressed. Specifically, system measurements and experimental benchmarks are required to understand the trade-off between migrating application tasks (or state) between edge servers and reducing latency towards the end users.

Security and Privacy: Security and privacy in video edge-C3 remains an open problem. In this context, securing both the processing and streaming of uplink and downlink video data is required. Securely caching and streaming downlink videos to end users has been well-studied, even in the context of edge-C3 (see [269] for a review of threat models and solutions). However, ensuring the security and privacy of uplink video streams raises several new challenges. First, secure processing of video frames (e.g., to detect objects) can be achieved using either homomorphic encryption and secure multiparty computation [129]. In homomorphic encryption, the input data is encrypted and analytics tasks are carried out directly on such encrypted data. However, this requires a large amount of computing resources. On the other hand, secure multiparty computation allows multiple servers to compute a function over the input data that is kept private. In the edge-C3, such an approach places stress on the communication resources as intermediate results need to be exchanged between the cooperating servers. An evaluation of such different approaches in the edge-C3 remains an open direction for future work. Specifically, it is important to quantify the impact of the above methods taking into account the limited computing resources of both UEs and edge servers, as well as the overhead in communication. Second, in the uplink, video streams may be manipulated to negatively impact video analytics tasks. This aspect is crucial for analytics based on crowdsourced video streams, but has not been well-studied in the literature. One approach to verify the integrity of the source is through watermarking the video frames, which can then be verified at the destination [270]. However, watermarking all frames is a compute-intensive process, whereas watermarking only certain key frames requires careful consideration (e.g., certain frames are more important from an analytics perspective). The design of an analytics pipeline that takes into account the integrity and security of processing while still maintaining a low latency (e.g., in the range of 100 ms for AR applications) is an open challenge. Finally, the privacy of end users in analytics systems is discussed only in a few articles [187], [188] that address such concerns in detecting faces (discussed in Section IV-G). However, several open challenges remain for general analytics tasks, where even input frames from a general environment may reveal private information of the end user. To this end, obfuscating the input data has been proposed to alleviate such concerns. Unfortunately, the amount of noise to be added may be large [129], as there may be only few users (and consequently less input data compared to a cloud-based solution) as well as specialized CNN models in the edge-C3. Thus, a rigorous analysis of the amount of noise for different analytics tasks and in the presence of different specialized models is required.

VII. Conclusion

This article presented a comprehensive review of video edge caching, computing, and communication (edge-C3) in next-generation wireless networks. In particular, it has first overviewed the core components of video streaming and how they can be extended to support emerging applications. Next, it has discussed the networking technologies for edge-C3 and the challenges associated with processing and delivering videos both in the uplink and the downlink. The latter part of the survey provided a thorough and up-to-date review of the state of the art in video edge-C3 according to different classes, based on the primary target of the considered solutions: the uplink (for video analytics at the edge) and the downlink (for edge-assisted video delivery). The works presented in each class have been crisply summarized, classified according to a novel taxonomy, and compared with each other. Several illustrations and summary tables therein further assist the reader in understanding the broad landscape of video edge-C3. Finally, the article provided insights on open issues and future research challenges in the considered context. We hope that this survey will help networking protocol designers and multimedia application developers to design efficient solutions for video streaming and delivery in future wireless networks.

References

[1] Cisco. (2019). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021*. [Online]. Available: https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1853168

[2] GSMA. (2019). *The Mobile Economy*. [Online]. Available: https://www.gsma.com/r/mobileeconomy/

[3] Y.-M. Hsiao, J.-F. Lee, J.-S. Chen, and Y.-S. Chu, "H.264 video transmissions over wireless networks: Challenges and solutions," *Comput. Commun.*, vol. 34, no. 14, pp. 1661–1672, 2011.

[4] S. Pudlewski and T. Melodia, "A tutorial on encoding and wireless transmission of compressively sampled videos," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 754–767, 2nd Quart., 2013.

[5] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, 1st Quart., 2015.

[6] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2847–2886, 4th Quart., 2016.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

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

[9] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[10] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[11] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[12] T. Zhao, Q. Liu, and C. W. Chen, "QoE in video transmission: A user experience-driven strategy," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 285–302, 1st Quart., 2017.

[13] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over HTTP," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 562–585, 1st Quart., 2019.

[14] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in information-centric networking: Strategies, challenges, and future research directions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1443–1474, 2nd Quart., 2018.

[15] L. Li, G. Zhao, and R. S. Blum, "A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1710–1732, 3rd Quart., 2018.

[16] C. Li, Y. Xue, J. Wang, W. Zhang, and T. Li, "Edge-oriented computing paradigms: A survey on architecture design and system management," *ACM Comput. Surveys*, vol. 51, no. 2, pp. 1–34, 2018.

[17] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. de Foy, and Y. Zhang, "Mobile edge cloud system: Architectures, challenges, and approaches," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2495–2508, Sep. 2018.

[18] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3098–3130, 4th Quart., 2018.

[19] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1111–1125, Jun. 2018.

[20] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, 4th Quart., 2018.

[21] M. T. Vega, C. Perra, F. D. Turck, and A. Liotta, "A review of predictive quality of experience management in video streaming services," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 432–445, Jun. 2018.

[22] C. Wang, Y. He, F. R. Yu, Q. Chen, and L. Tang, "Integration of networking, caching, and computing in wireless systems: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 7–38, 1st Quart., 2018.

[23] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2525–2553, 3rd Quart., 2019.

[24] A. A. Barakabitze *et al.*, "QoE management of multimedia streaming services in future networks: A tutorial and survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 526–565, 1st Quart., 2020.

[25] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[26] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[27] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[28] Y. C. Hu, Y. C. Hu, Y. C. Hu, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," Eur. Telecommun. Standards Inst., Sophia Antipolis, France, White Paper, 2015.

[29] Amazon Web Services. *AWS Wavelength*. Accessed: Nov. 23, 2020. [Online]. Available: https://aws.amazon.com/wavelength/

[30] Microsoft Azure. *About Azure Edge Zone Preview*. Accessed: Nov. 23, 2020. [Online]. Available: https://docs.microsoft.com/en-us/azure/networking/edge-zones-overview

[31] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1473–1499, 3rd Quart., 2015.

[32] Y. Wang and X. Lin, "CHetNet: Crowdsourcing to heterogeneous cellular networks," *IEEE Netw.*, vol. 29, no. 6, pp. 62–67, Nov./Dec. 2015.

[33] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8095–8113, Oct. 2019.

[34] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1801–1819, 4th Quart., 2014.

[35] B. Jedari, F. Xia, and Z. Ning, "A survey on human-centric communications in non-cooperative wireless relay networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 914–944, 2nd Quart., 2018.

[36] M. Zink, R. Sitaraman, and K. Nahrstedt, "Scalable 360° video stream delivery: Challenges, solutions, and opportunities," *Proc. IEEE*, vol. 107, no. 4, pp. 639–650, Apr. 2019.

[37] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas, "Video delivery over heterogeneous cellular networks: Optimizing cost and performance," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2014, pp. 1078–1086.

[38] A. Begen, T. Akgul, and M. Baugher, "Watching video over the Web: Part 1: Streaming protocols," *IEEE Internet Comput.*, vol. 15, no. 2, pp. 54–63, Mar./Apr. 2010.

[39] C. Zhu, Y. Li, and X. Niu, *Streaming Media Architectures, Techniques, and Applications: Recent Advances*. Hershey, PA, USA: IGI Global, 2011.

[40] A. Begen, T. Akgul, and M. Baugher, "Watching video over the Web: Part 2: Applications, standardization, and open issues," *IEEE Internet Comput.*, vol. 15, no. 3, pp. 59–63, May/Jun. 2010.

[41] B. Wandell and S. Thomas, "Foundations of vision," *Psyccritiques*, vol. 42, no. 7, p. 649, 1997.

[42] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, pp. 34–47, Nov./Dec. 2001.

[43] S. Notebaert, J. De Cock, S. Beheydt, J. De Lameillieure, and R. Van de Walle, "Mixed architectures for H.264/AVC digital video transrating," *Multimedia Tools Appl.*, vol. 44, no. 1, pp. 39–64, 2009.

[44] A. Tripathi and M. Claypool, "Improving multimedia streaming with content-aware video scaling," Comput. Sci. Dept., Worcester Polytech. Inst., Worcester, MA, USA, Rep. WPI-CS-TR-01-02, 2001. [Online]. Available: https://digitalcommons.wpi.edu/computerscience-pubs/96/

[45] "Subjective video quality assessment methods for multimedia applications," Int. Telecommun. Union, Geneva, Switzerland, ITU-Recommendation P. 910, Apr. 2008. [Online]. Available: https://www.itu.int/rec/T-REC-P. 910-200804-I

[46] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010.

[47] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara. (2016). *Toward a Practical Perceptual Video Quality Metric*. Accessed: Nov. 23, 2020. [Online]. Available: https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

[48] T. Kämäräinen, M. Siekkinen, A. Ylä-Jääski, W. Zhang, and P. Hui, "A measurement study on achieving imperceptible latency in mobile cloud gaming," in *Proc. 8th ACM Multimedia Syst. Conf.* 2017, pp. 88–99.

[49] T. Kämäräinen, M. Siekkinen, J. Eerikäinen, and A. Ylä-Jääski, "CloudVR: Cloud accelerated interactive mobile virtual reality," in *Proc. ACM Multimedia Conf.*. 2018, pp. 1181–1189.

[50] *High Definition (HD) Image Formats for Television Production*, document EBU-Tech 3299, Eur. Broadcast. Union, Geneva, Switzerland, 2004.

[51] "Parameter values for ultra-high definition television systems for production and international programme exchange," Int. Telecommun. Union, Geneva, Switzerland, ITU-Recommendation BT. 2020-2, Oct. 2015. [Online]. Available: https://www.itu.int/rec/R-REC-BT. 2020-2-201510-I

[52] R. Schatz, A. Sackl, C. Timmerer, and B. Gardlo, "Towards subjective quality of experience assessment for omnidirectional video streaming," in *Proc. 9th Int. Conf. Qual. Multimedia Experience (QoMEX)*, May 2017, pp. 1–6.

[53] A. C. Bovik, *Handbook of Image and Video Processing*. Boston, MA, USA: Academic Press, 2010.

[54] Encoding.Com. (2018). *Global Media Formats Report*. [Online]. Available: https://www.encoding.com/files/2018-Global-Media-Formats-Report.pdf

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

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

[57] B. Bross, J. Chen, and S. Liu, *Versatile Video Coding (Draft 1)*, document JVET-J1001, Joint Video Exploration Team (JVET), San Diego, CA, USA, 2018.

[58] D. Mukherjee *et al.*, "The latest open-source video codec VP9—An overview and preliminary results," in *Proc. IEEE Pict. Coding Symp. (PCS)*, 2013, pp. 390–393.

[59] Y. Chen *et al.*, "An overview of core coding tools in the AV1 video codec," in *Proc. IEEE Pict. Coding Symp. (PCS)*, 2018, pp. 41–45.

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

[61] Cisco. (2014). *Emerging Video Technologies: H.265, SVC, and WebRTC*. [Online]. Available: https://www.ciscolive.com

[62] C. Zhan and Z. Wen, "Content cache placement for scalable video in heterogeneous wireless network," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2714–2717, Dec. 2017.

[63] L. Wu and W. Zhang, "Caching-based scalable video transmission over cellular networks," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1156–1159, Jun. 2016.

[64] Q. Jiang, V. C. M. Leung, H. Tang, and H.-S. Xi, "Energy-efficient traffic rate adaptation for wireless streaming media transmission," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3313–3319, Nov. 2018.

[65] R. Yu *et al.*, "Enhancing software-defined RAN with collaborative caching and scalable video coding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2016, pp. 1–6.

[66] T. Schierl, C. Hellge, S. Mirta, K. Gruneberg, and T. Wiegand, "Using H.264/AVC-based scalable video coding (SVC) for real time streaming in wireless IP networks," in *Proc. IEEE Int. Symp. Circuits Syst.*. 2007, pp. 3455–3458.

[67] C. Pakha, A. Chowdhery, and J. Jiang, "Reinventing video streaming for distributed vision analytics," in *Proc. 10th USENIX Workshop Hot Topics Cloud Comput. (HotCloud)*, 2018, p. 1.

[68] T. Stockhammer, "Dynamic adaptive streaming over HTTP –: Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, New York, USA, 2011, pp. 133–144.

[69] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," Internet Soc., Columbia Univ., Vancouver, BC, Canada, Rep. RFC 2326, 1998.

[70] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," Internet Eng. Task Force, Fremont, CA, USA, RFC 3550, 2003.

[71] H. Parmar and M. Thornburgh, "Adobe's real time messaging protocol," Adobe Inc., San Jose, CA, USA, Rep., 2012. [Online]. Available: https://www.adobe.com/devnet/rtmp.html

[72] J. K. Nurminen, A. J. R. Meyn, E. Jalonen, Y. Raivio, and R. G. Marrero, "P2P media streaming with HTML5 and WebRTC," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2013, pp. 63–64.

[73] M. Siekkinen, E. Masala, and T. Kämäräinen, "A first look at quality of mobile live streaming experience: The case of Periscope," in *Proc. Internet Meas. Conf.*, 2016, pp. 477–483.

[74] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[75] V. K. Adhikari, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling Netflix: Understanding and improving multi-CDN movie delivery," in *Proc. IEEE INFOCOM*, 2012, pp. 1620–1628.

[76] P. Casas, A. D'Alconzo, P. Fiadino, A. Bär, A. Finamore, and T. Zseby, "When YouTube does not work—Analysis of QoE-relevant degradation in Google CDN traffic," *IEEE Trans. Netw. Service Manag.*, vol. 11, no. 4, pp. 441–457, Dec. 2014.

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

[78] G. Cofano, L. D. Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo, "Design and performance evaluation of network-assisted control strategies for HTTP adaptive streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 3s, pp. 1–24, 2017.

[79] A. Heikkinen, "Network-assisted DASH by utilizing local caches at network edge," in *Proc. 26th IEEE Int. Conf. Softw. Telecommun. Comput. Netw. (SoftCOM)*, 2018, pp. 1–6.

[80] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Edge computing assisted adaptive mobile video streaming," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 787–800, Apr. 2019.

[81] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 99–114.

[82] M. Xiao, C. Zhou, V. Swaminathan, Y. Liu, and S. Chen, "BAS-360°: Exploring spatial and temporal adaptability in 360-degree videos over HTTP/2," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2018, pp. 953–961.

[83] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2020, pp. 1977–1986.

[84] S. Shi, V. Gupta, and R. Jana, "Freedom: Fast recovery enhanced VR delivery over mobile networks," in *Proc. 17th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2019, pp. 130–141.

[85] R. Stankiewicz, P. Cholda, and A. Jajszczyk, "QoX: What is it really?" *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 148–158, Apr. 2011.

[86] "Definitions of terms related to quality of service," Int. Telecommun. Union, Geneva, Switzerland, ITU-Recommendation E.800, Sep. 2008. [Online]. Available: https://www.itu.int/rec/T-REC-E.800-200809-I

[87] "Vocabulary for performance, quality of service and quality of experience," Int. Telecommun. Union, Geneva, Switzerland, ITU-Recommendation P. 10/G.100, Nov. 2017. [Online]. Available: https://www.itu.int/rec/T-REC-P. 10-201711-I

[88] C. Keighrey, R. Flynn, S. Murray, and N. Murray, "A QoE evaluation of immersive augmented and virtual reality speech & language assessment applications," in *Proc. 9th Int. Conf. Quality Multimedia Experience (QoMEX)*, 2017, pp. 1–6.

[89] K. Brunnström *et al.*, "Qualinet white paper on definitions of quality of experience," IEEE, Erfurt, Germany, White Paper, 2013.

[90] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of quality of experience of video-on-demand services: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 401–418, 1st Quart., 2016.

[91] "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport," Int. Telecommun. Union, Geneva, Switzerland , ITU-Recommendation P. 1203, Oct. 2017. [Online]. Available: https://www.itu.int/rec/T-REC-P. 1203-201710-I

[92] "Video quality assessment of streaming services over reliable transport for resolutions up to 4K," Int. Telecommun. Union, Geneva, Switzerland, ITU-Recommendation P. 1204, Jan. 2020. [Online]. Available: https://www.itu.int/rec/T-REC-P. 1204-202001-I

[93] M. Yang, S. Wang, R. N. Calheiros, and F. Yang, "Survey on QoE assessment approach for network service," *IEEE Access*, vol. 6, pp. 48374–48390, 2018.

[94] C. Liang, Y. He, F. R. Yu, and N. Zhao, "Enhancing video rate adaptation with mobile edge computing and caching in software-defined mobile networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 7013–7026, Oct. 2018.

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

[96] J. Wu, B. Cheng, Y. Yang, M. Wang, and J. Chen, "Delay-aware quality optimization in cloud-assisted video streaming system," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 1, pp. 1–25, 2017.

[97] A. Khan, L. Sun, and E. Ifeachor, "QoE prediction model and its application in video quality adaptation over UMTS networks," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 431–442, Apr. 2012.

[98] J. Nightingale, P. Salva-Garcia, J. M. A. Calero, and Q. Wang, "5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 621–634, Jun. 2018.

[99] C. Ge and N. Wang, "Real-time QoE estimation of DASH-based mobile video applications through edge computing," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2018, pp. 766–771.

[100] C. Tselios and G. Tsolis, "On QoE-awareness through virtualized probes in 5G networks," in *Proc. IEEE 21st Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, 2016, pp. 159–164.

[101] S.-R. Yang, Y.-J. Tseng, C.-C. Huang, and W.-C. Lin, "Multi-access edge computing enhanced video streaming: Proof-of-concept implementation and prediction/QoE models," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1888–1902, Feb. 2019.

[102] D. Tsolkas, E. Liotou, N. Passas, and L. Merakos, "A survey on parametric QoE estimation for popular services," *J. Netw. Comput. Appl.*, vol. 77, pp. 1–17, Jan. 2017.

[103] K. D. Singh, Y. Hadjadj-Aoul, and G. Rubino, "Quality of experience estimation for adaptive HTTP/TCP video streaming using H.264/AVC," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, 2012, pp. 127–131.

[104] Y. Li, P. A. Frangoudis, Y. Hadjadj-Aoul, and P. Bertin, "A mobile edge computing-based architecture for improved adaptive HTTP video delivery," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, 2016, pp. 1–6.

[105] "Technical specification group services and system aspects; Release 15 description; Summary of Rel-15 work items, version 15.0.0," 3GPP, Sophia Antipolis, France, Rep. (TR) 21.915, Sep. 2019. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3389

[106] "5G; Service requirements for next generation new services and markets, version 15.5.0," 3GPP, Sophia Antipolis, France, Rep. (TS) 22.261, Apr. 2018, [Online]. Available: https://www.etsi.org/deliver/etsi_ts/122200_122299/122261/15.05.00_60/ts_122261v150500p.pdf

[107] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3GPP NR at mmWave frequencies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 173–196, 1st Quart., 2018.

[108] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive MIMO is a reality—What is next?: Five promising research directions for antenna arrays," *Digit. Signal Process.*, vol. 94, pp. 3–20, Nov. 2019.

[109] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[110] N. Bhushan *et al.*, "Network densification: The dominant theme for wireless evolution into 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 82–89, Feb. 2014.

[111] M. Minowa *et al.*, "5G R&D activities for high capacity technologies with ultra high-density multi-band and multi-access layered cells," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*. 2019, pp. 1–5.

[112] C. Saha and H. S. Dhillon, "Millimeter wave integrated access and Backhaul in 5G: Performance analysis and design insights," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 12, pp. 2669–2684, Dec. 2019.

[113] C. Wang, R. C. Elliott, D. Feng, W. A. Krzymien, S. Zhang, and J. Melzer, "A Framework for MEC-enhanced small-cell HetNet with massive MIMO," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 64–72, Aug. 2020.

[114] F. S. Shaikh and R. Wismüller, "Routing in multi-hop cellular device-to-device (D2D) networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2622–2657, 4th Quart., 2018.

[115] R. I. Ansari *et al.*, "5G D2D networks: Techniques, challenges, and future prospects," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3970–3984, Dec. 2018.

[116] G. Ding *et al.*, "Spectrum inference in cognitive radio networks: Algorithms and applications," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 150–182, 1st Quart., 2018.

[117] M. Amjad, M. H. Rehmani, and S. Mao, "Wireless multimedia cognitive radio networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1056–1103, 2nd Quart., 2018.

[118] Z. He, S. Mao, and T. Jiang, "A survey of QoE-driven video streaming over cognitive radio networks," *IEEE Netw.*, vol. 29, no. 6, pp. 20–25, Nov./Dec. 2015.

[119] C. Xin and M. Song, "Analysis of the on-demand spectrum access architecture for CBRS cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 970–978, Feb. 2019.

[120] K. Mun, "OnGo: New shared spectrum enables flexible indoor and outdoor mobile solutions and new business models," Mobile Experts Inc., Campbell, CA, USA, Rep., 2018. [Online]. Available: https://www.cbrsalliance.org/wp-content/uploads/2018/04/Mobile-/

[121] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[122] I. T. Haque and N. Abu-Ghazaleh, "Wireless software defined networking: A survey and taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2713–2737, 4th Quart., 2016.

[123] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 146–172, 1st Quart., 2019.

[124] The Linux Foundation. (2019). *Open Glossary of Edge Computing, Version 2.0*. [Online]. Available: https://github.com/State-of-the-Edge/glossary

[125] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[126] F. Giust *et al.*, "MEC deployments in 4G and evolution towards 5G," ETSI, Sophia Antipolis, France, White Paper, 2018.

[127] S. Kekki *et al.*, "MEC in 5G networks," ETSI, Sophia Antipolis, France, White Paper, 2018.

[128] M. C. Filippou *et al.*, "Multi-access edge computing: A comparative analysis of 5G system deployments and service consumption locality variants," *IEEE Commun. Stand. Mag.*, vol. 4, no. 2, pp. 32–39, Jun. 2020.

[129] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[130] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, "Femto clouds: Leveraging mobile devices to provide cloud service at the edge," in *Proc. IEEE 8th Int. Conf. Cloud Comput. (CLOUD)*. 2015, pp. 9–16.

[131] G. Tang, H. Wang, K. Wu, and D. Guo, "Tapping the knowledge of dynamic traffic demands for optimal CDN design," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 98–111, Feb. 2019.

[132] D. De Vleeschauwer and D. C. Robinson, "Optimum caching strategies for a telco CDN," *Bell Labs Tech. J.*, vol. 16, no. 2, pp. 115–132, 2011.

[133] A. Martin, R. Viola, M. Zorrilla, J. Flórez, P. Angueira, and J. Montalbán, "MEC for fair, reliable and efficient media streaming in mobile networks," *IEEE Trans. Broadcast.*, vol. 66, no. 2, pp. 264–278, Jun. 2020.

[134] Akraino. *Akraino Edge Stack*. Accessed: Nov. 23, 2020. [Online]. Available: https://wiki.akraino.org/display/AK/Akraino+Edge+Stack

[135] The Linux Foundation. *EdgeXFoundry Documentation*. Accessed: Nov. 23, 2020. [Online]. Available: https://docs.edgexfoundry.org/1.2/

[136] Open Networking Foundation. *CORD*. Accessed: Nov. 23, 2020. [Online]. Available: https://www.opennetworking.org/cord/

[137] L. Peterson *et al.*, "Central office re-architected as a data center," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 96–101, Oct. 2016.

[138] Open Networking Foundation. *Aether*. Accessed: Nov. 23, 2020. [Online]. Available: https://www.opennetworking.org/aether/

[139] J. Kim, Y. Jung, H. Yeo, J. Ye, and D. Han, "Neural-enhanced live streaming: Improving live video ingest via online learning," in *Proc. Annu. Conf. ACM Spec. Interest Group Data Commun. Appl. Technol. Arch. Protocols Comput. Commun.*, 2020, pp. 107–125.

[140] G. Ananthanarayanan *et al.*, "Real-time video analytics: The killer app for edge computing," *Computer*, vol. 50, no. 10, pp. 58–67, Oct. 2017.

[141] K. Du *et al.*, "Server-driven video streaming for deep learning inference," in *Proc. Annu. Conf. ACM Spec. Interest Group Data Commun. Appl. Technol. Arch. Protocols Comput. Commun.*, 2020, pp. 557–570.

[142] H. Trinh *et al.*, "Energy-aware mobile edge computing and routing for low-latency visual data processing," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2562–2577, Oct. 2018.

[143] X. Ran, H. Chen, X. Zhu, Z. Liu, and J. Chen, "DeepDecision: A mobile deep learning framework for edge video analytics," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 1421–1429.

[144] H. Ding, Y. Guo, X. Li, and Y. Fang, "Beef Up the Edge: Spectrum-aware placement of edge computing services for the Internet of things," *IEEE Trans. Mobile Comput.*, vol. 18, no. 12, pp. 2783–2795, Dec. 2019.

[145] W. Zhang, B. Han, and P. Hui, "Jaguar: Low latency mobile augmented reality with flexible tracking," in *Proc. ACM Multimedia Conf.*, 2018, pp. 355–363.

[146] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–16.

[147] P. Loncomilla, J. Ruiz-del Solar, and L. Martínez, "Object recognition using local invariant features for robotic applications: A survey," *Pattern Recognit.*, vol. 60, pp. 499–514, Dec. 2016.

[148] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.

[149] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Computer Vision (ECCV)*. Berlin, Germany: Springer, 2006, pp. 404–417.

[150] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[151] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2005, pp. 886–893.

[152] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[153] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: A survey," *Pattern Recognit.*, vol. 51, pp. 148–175, Mar. 2016.

[154] U. Drolia, K. Guo, J. Tan, R. Gandhi, and P. Narasimhan, "Cachier: Edge-caching for recognition applications," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2017, pp. 276–286.

[155] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[156] S. Pouyanfar *et al.*, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surveys*, vol. 51, no. 5, pp. 1–36, 2018.

[157] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: arXiv:1409.1556.

[158] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[159] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[160] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[161] U. Drolia, K. Guo, and P. Narasimhan, "Precog: Prefetching for image recognition applications at the edge," in *Proc. 2nd ACM IEEE Symp. Edge Comput. (SEC)*, 2017, pp. 1–13.

[162] H. Li, K. Ota, and M. Dong, "Learning IoT in Edge: Deep learning for the Internet of things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan./Feb. 2018.

[163] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman, "Live video analytics at scale with approximation and delay-tolerance," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implementation*, 2017, pp. 377–392.

[164] C.-C. Hung *et al.*, "VideoEdge: Processing Camera streams using hierarchical clusters," in *Proc. IEEE ACM Symp. Edge Comput. (SEC)*, 2018, pp. 115–131.

[165] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, and Q. Li, "LAVEA: Latency-aware video analytics on edge computing platform," in *Proc. 2nd IEEE ACM Symp. Edge Comput.*, 2017, pp. 1–13.

[166] Z. Lu, K. S. Chan, and T. L. Porta, "A computing platform for video crowdprocessing using deep learning," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 1430–1438.

[167] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Services*. 2010, pp. 49–62.

[168] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1126–1139, May 2018.

[169] T. Zhang, A. Chowdhery, P. V. Bahl, K. Jamieson, and S. Banerjee, "The design and implementation of a wireless video surveillance system," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2015, pp. 426–438.

[170] S. Y. Jang, Y. Lee, B. Shin, and D. Lee, "Application-aware IoT camera virtualization for video analytics edge computing," in *Proc. IEEE ACM Symp. Edge Comput. (SEC)*, 2018, pp. 132–144.

[171] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.

[172] T. Y.-H. Chen, L. Ravindranath, S. Deng, P. Bahl, and H. Balakrishnan, "Glimpse: Continuous, real-time object recognition on mobile devices," in *Proc. 13th ACM Conf. Embedded Netw. Sens. Syst. SenSys*, 2015, pp. 155–168.

[173] S. Han, H. Shen, M. Philipose, S. Agarwal, A. Wolman, and A. Krishnamurthy, "MCDNN: An approximation-based execution framework for deep stream processing under resource constraints," in *Proc. 14th Annu. Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, 2016, pp. 123–136.

[174] Q. Liu, S. Huang, J. Opadere, and T. Han, "An edge network orchestrator for mobile augmented reality," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 756–764.

[175] S. M. Azimi, "ShuffleDet: Real-time vehicle detection network in on-board embedded UAV imagery," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 88–99.

[176] N. Tijtgat, W. Van Ranst, T. Goedeme, B. Volckaert, and F. De Turck, "Embedded real-time object detection for a UAV warning system," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2110–2118.

[177] S. Mittal, "A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform," *J. Syst. Arch.*, vol. 97, pp. 428–442, Aug. 2019.

[178] J. Wang *et al.*, "Bandwidth-efficient live video analytics for drones via edge computing," in *Proc. IEEE ACM Symp. Edge Comput. (SEC)*, 2018, pp. 159–173.

[179] A. Chowdhery and M. Chiang, "Model predictive compression for drone video analytics," in *Proc. IEEE Int. Conf. Sens. Commun. Netw. (SECON Workshops)*, 2018, pp. 1–5.

[180] X. Wang, A. Chowdhery, and M. Chiang, "Networked drone cameras for sports streaming," in *Proc. 37th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2017, pp. 308–318.

[181] *OpenALPR—Automatic License Plate Recognition*. Accessed: Apr. 2, 2019. [Online]. Available: https://www.openalpr.com/

[182] Q. Zhang *et al.*, "OpenVDAP: An open vehicular data analytics platform for CAVs," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 1310–1320.

[183] Q. Zhang, Q. Zhang, W. Shi, and H. Zhong, "Firework: Data processing and sharing for hybrid cloud-edge analytics," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 9, pp. 2004–2017, Sep. 2018.

[184] H. Qiu *et al.*, "Kestrel: Video analytics for augmented multi-camera vehicle tracking," in *Proc. IEEE/ACM 3rd Int. Conf. Internet Things Design Implementation (IoTDI)*. 2018, pp. 48–59.

[185] G. Grassi, K. Jamieson, P. Bahl, and G. Pau, "Parkmaster: An in-vehicle, edge-based video analytics service for detecting open parking spaces in urban environments," in *Proc. 2nd ACM/IEEE Symp. Edge. Comput. SEC*, 2017, pp. 1–14.

[186] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, and M. Satyanarayanan, "Scalable crowd-sourcing of video from mobile devices," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, 2013, pp. 139–152.

[187] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, "Enabling live video analytics with a scalable and privacy-aware framework," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 3s, pp. 1–24, 2018.

[188] S. A. Miraftabzadeh, P. Rad, K.-K. R. Choo, and M. Jamshidi, "A privacy-aware architecture at the edge for autonomous real-time identity reidentification in crowds," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2936–2946, Aug. 2018.

[189] P. Dogga, S. Chakraborty, S. Mitra, and R. Netravali, "Edge-based transcoding for adaptive live video streaming," in *Proc. 2nd USENIX Workshop Hot Topics Edge Comput. (HotEdge)*. Jul. 2019, pp. 1–7.

[190] M. Ma *et al.*, "Characterizing user behaviors in mobile personal livecast: Towards an edge computing-assisted paradigm," *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)*, vol. 14, no. 3s, pp. 1–24, 2018.

[191] A. Raman, G. Tyson, and N. Sastry, "Facebook (A)Live? Are live social broadcasts really broadcasts?" in *Proc. World Wide Web Conf.*, 2018, pp. 1491–1500.

[192] J. Chen, B. Balasubramanian, and Z. Huang, "Liv(e)-ing on the edge: User-uploaded live streams driven by "first-mile" edge decisions," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, 2019, pp. 41–50.

[193] Y. Zhu, Q. He, J. Liu, B. Li, and Y. Hu, "When crowd meets big video data: Cloud-edge collaborative transcoding for personal livecast," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 42–53, Jan./Mar. 2020.

[194] S. Naderiparizi, P. Zhang, M. Philipose, B. Priyantha, J. Liu, and D. Ganesan, "Glimpse: A programmable early-discard camera architecture for continuous mobile vision," in *Proc. 15th Annu. Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, 2017, pp. 292–305.

[195] M. Chen, Y. Hao, L. Hu, M. S. Hossain, and A. Ghoneim, "Edge-CoCaCo: Toward joint optimization of computation, caching, and communication on edge cloud," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 21–27, Jun. 2018.

[196] D. Wang *et al.*, "Adaptive wireless video streaming based on edge computing: Opportunities and approaches," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 685–697, Sep./Oct. 2019.

[197] M. Tang, L. Gao, H. Pang, J. Huang, and L. Sun, "Optimizations and economics of crowdsourced mobile streaming," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 21–27, Apr. 2017.

[198] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[199] T. X. Tran and D. Pompili, "Octopus: A cooperative hierarchical caching strategy for cloud radio access networks," in *Proc. 13th Int. Conf. Mobile Ad Hoc Sens. Syst. (MASS)*, 2016, pp. 154–162.

[200] Z. Qu, B. Ye, B. Tang, S. Guo, S. Lu, and W. Zhuang, "Cooperative caching for multiple bitrate videos in small cell edges," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 288–299, Feb. 2020.

[201] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[202] W. C. Ao and K. Psounis, "Fast content delivery via distributed caching and small cell cooperation," *IEEE Trans. Mobile Comput.*, vol. 17, no. 5, pp. 1048–1061, May 2018.

[203] X. Liu, N. Zhao, F. R. Yu, Y. Chen, J. Tang, and V. C. M. Leung, "Cooperative video transmission strategies via caching in small-cell networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12204–12217, Dec. 2018.

[204] P. Maniotis, E. Bourtsoulatze, and N. Thomos, "Tile-based joint caching and delivery of 360° videos in heterogeneous networks," *IEEE Trans. Multimedia*, vol. 22, no. 9, pp. 2382–2395, Sep. 2019.

[205] J. Dai, Z. Zhang, S. Mao, and D. Liu, "A view synthesis-based 360° VR caching system over MEC-enabled C-RAN," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3843–3855, Oct. 2020.

[206] L. Sun, M. Ma, W. Hu, H. Pang, and Z. Wang, "Beyond 1 million nodes: A crowdsourced video content delivery network," *IEEE MultiMedia*, vol. 24, no. 3, pp. 54–63, Aug. 2017.

[207] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[208] L. Zhou, "Mobile device-to-device video distribution: Theory and application," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 3, pp. 1–23, 2016.

[209] J. Kim, G. Caire, and A. F. Molisch, "Quality-aware streaming and scheduling for device-to-device video delivery," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2319–2331, Aug. 2016.

[210] D. Wu, J. Huang, J. He, M. Chen, and G. Zhang, "Toward cost-effective mobile video streaming via smart cache with adaptive thresholding," *IEEE Trans. Broadcast.*, vol. 61, no. 4, pp. 639–650, Dec. 2015.

[211] X. Zhang, D. Xiong, K. Zhao, C. W. Chen, and T. Zhang, "Realizing low-cost flash memory based video caching in content delivery systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 4, pp. 984–996, Apr. 2018.

[212] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[213] P. Pirozmand, G. Wu, B. Jedari, and F. Xia, "Human mobility in opportunistic networks: Characteristics, models and prediction methods," *J. Netw. Comput. Appl.*, vol. 42, pp. 45–58, Jun. 2014.

[214] A. Tatar, M. D. De Amorim, S. Fdida, and P. Antoniadis, "A survey on predicting the popularity of Web content," *J. Internet Services Appl.*, vol. 5, no. 1, p. 8, 2014.

[215] T. Hou, G. Feng, S. Qin, and W. Jiang, "Proactive content caching by exploiting transfer learning for mobile edge computing," *Int. J. Commun. Syst.*, vol. 31, no. 11, 2018, Art. no. e3706.

[216] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1024–1036, Feb. 2017.

[217] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3520–3535, Jun. 2017.

[218] W. Li, S. M. A. Oteafy, and H. S. Hassanein, "StreamCache: Popularity-based caching for adaptive streaming over information-centric networks," in *Proc. IEEE Int. Conf. Commun. (IEEE ICC)*, 2016, pp. 1–6.

[219] W. Hoiles, O. N. Gharehshiran, V. Krishnamurthy, N.-D. Dao, and H. Zhang, "Adaptive caching in the youtube content distribution network: A revealed preference game-theoretic learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 1, no. 1, pp. 71–85, Mar. 2015.

[220] J. Liu, H. Yan, Y. Li, D. Wu, L. Su, and D. Jin, "Cache behavior characterization and validation over large-scale video data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 734–745, Mar. 2018.

[221] N. Carlsson and D. Eager, "Ephemeral content popularity at the edge and implications for on-demand caching," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 6, pp. 1621–1634, Jun. 2017.

[222] J.-P. Hong and W. Choi, "User prefix caching for average playback delay reduction in wireless video streaming," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 377–388, Jan. 2016.

[223] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[224] K. Mokhtarian and H.-A. Jacobsen, "Flexible caching algorithms for video content distribution networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1062–1075, Apr. 2017.

[225] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1444–1462, Oct. 2014.

[226] J. Qiao, Y. He, and X. S. Shen, "Proactive caching for mobile video streaming in millimeter wave 5G networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 7187–7198, Oct. 2016.

[227] M. Claeys, N. Bouten, D. De Vleeschauwer, W. Van Leekwijck, S. Latre, and F. De Turck, "Cooperative announcement-based caching for video-on-demand streaming," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 2, pp. 308–321, Jun. 2016.

[228] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 80–87, Jun. 2018.

[229] Y. Guo, Q. Yang, F. R. Yu, and V. C. M. Leung, "Cache-enabled adaptive video streaming over vehicular networks: A dynamic approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5445–5459, Jun. 2018.

[230] J. Dai, Z. Zhang, and D. Liu, "Proactive caching over cloud radio access network with user mobility and video segment popularity awared," *IEEE Access*, vol. 6, pp. 44396–44405, 2018.

[231] N. Kumar, S. Zeadally, and J. J. P. C. Rodrigues, "QoS-aware hierarchical web caching scheme for online video streaming applications in Internet-based vehicular ad hoc networks," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7892–7900, Dec. 2015.

[232] L. Vigneri, T. Spyropoulos, and C. Barakat, "Low cost video streaming through mobile edge caching: Modelling and optimization," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6. pp. 1302–1315, Jun. 2019.

[233] J. Li, Z. Ning, B. Jedari, F. Xia, I. Lee, and A. Tolba, "Geo-social distance-based data dissemination for socially aware networking," *IEEE Access*, vol. 4, pp. 1444–1453, 2016.

[234] Z. Su, Q. Xu, F. Hou, Q. Yang, and Q. Qi, "Edge caching for layered video contents in mobile social networks," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2210–2221, Oct. 2017.

[235] J. Xu, M. van der Schaar, J. Liu, and H. Li, "Forecasting popularity of videos using social media," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 2, pp. 330–343, Mar. 2015.

[236] D. Wu, J. Yan, H. Wang, D. Wu, and R. Wang, "Social attribute aware incentive mechanism for device-to-device video distribution," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1908–1920, Aug. 2017.

[237] X. Zhao, P. Yuan, H. Li, and S. Tang, "Collaborative edge caching in context-aware device-to-device networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9583–9596, Oct. 2018.

[238] P. Sermpezis, T. Spyropoulos, L. Vigneri, and T. Giannakas, "Femto-caching with soft cache hits: Improving performance with related content recommendation," in *Proc. IEEE Glob. Commun. Conf.*, 2017, pp. 1–7.

[239] B. A. Wandell, *Foundations of Vision*. Sunderland, MA, USA: Sinauer Associates, 1995.

[240] J. Ryoo, K. Yun, D. Samaras, S. R. Das, and G. Zelinsky, "Design and evaluation of a foveated video streaming service for commodity client devices," in *Proc. 7th Int. Conf. Multimedia Syst.*, 2016, pp. 1–11.

[241] G. K. Illahi, T. V. Gemert, M. Siekkinen, E. Masala, A. Oulasvirta, and A. Ylä-Jääski, "Cloud gaming with foveated video encoding," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 16, no. 1, pp. 1–24, 2020.

[242] A. Mahzari, A. T. Nasrabadi, A. Samiei, and R. Prakash, "FoV-aware edge caching for adaptive 360° video streaming," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 173–181.

[243] G. Papaioannou and I. Koutsopoulos, "Tile-based caching optimization for 360° videos," in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2019, pp. 171–180.

[244] C. Perfecto, M. S. Elbamby, J. Del Ser, and M. Bennis, "Taming the latency in multi-user VR 360°: A QoE-aware deep learning-aided multicast framework," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2491–2508, Apr. 2020.

[245] P. Lungaro, R. Sjöberg, A. J. F. Valero, A. Mittal, and K. Tollmar, "Gaze-aware streaming solutions for the next generation of mobile VR experiences," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 4, pp. 1535–1544, Apr. 2018.

[246] Y. Jin, Y. Wen, and C. Westphal, "Optimal transcoding and caching for adaptive streaming in media cloud: An analytical approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 1914–1925, Dec. 2015.

[247] X. Xu, J. Liu, and X. Tao, "Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation," *IEEE Access*, vol. 5, pp. 16406–16415, 2017.

[248] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "QoE-traffic optimization through collaborative edge caching in adaptive mobile video streaming," *IEEE Access*, vol. 6, pp. 52261–52276, 2018.

[249] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "MEC-assisted panoramic VR video streaming over millimeter wave mobile networks," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1302–1316, May 2019.

[250] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communications, caching, and computing for mobile virtual reality: Modeling and tradeoff," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7573–7586, Nov. 2019.

[251] C. Ge, N. Wang, W. K. Chai, and H. Hellwagner, "QoE-assured 4K HTTP live streaming via transient segment holding at mobile edge," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1816–1830, Aug. 2018.

[252] Z. Zhang, R. Wang, F. R. Yu, F. Fu, and Q. Yan, "Qos aware transcoding for live streaming in edge-clouds aided hetnets: An enhanced actor-critic approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11295–11308, Nov. 2019.

[253] Y.-H. Hung, C.-Y. Wang, and R.-H. Hwang, "Optimizing social welfare of live video streaming services in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 4, pp. 922–934, Apr. 2020.

[254] S. E. Ghoreishi, V. Friderikos, D. Karamshuk, N. Sastry, and A. H. Aghvami, "Provisioning cost-effective mobile video caching," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2016, pp. 1–7.

[255] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Video transcoding, caching, and multicast for heterogeneous networks over wireless network virtualization," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 141–144, Jan. 2018.

[256] L. Sun, H. Pang, and L. Gao, "Joint sponsor scheduling in cellular and edge caching networks for mobile video delivery," *IEEE Trans. Multimedia*, vol. 20, no. 12, pp. 3414–3427, Dec. 2018.

[257] J. Li, W. Chen, M. Xiao, F. Shu, and X. Liu, "Efficient video pricing and caching in heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8744–8751, Oct. 2016.

[258] J. Li, J. Sun, Y. Qian, F. Shu, M. Xiao, and W. Xiang, "A commercial video-caching system for small-cell cellular networks using game theory," *IEEE Access*, vol. 4, pp. 7519–7531, 2016.

[259] J. Li, H. Chen, Y. Chen, Z. Lin, B. Vucetic, and L. Hanzo, "Pricing and resource allocation via game theory for a small-cell video caching system," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2115–2129, Aug. 2016.

[260] J. Dai, F. Liu, B. Li, B. Li, and J. Liu, "Collaborative caching in wireless video streaming through resource auctions," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 458–466, Feb. 2012.

[261] B. Jedari and M. D. Francesco, "Auction-based cache trading for scalable videos in multi-provider heterogeneous networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2019, pp. 1864–1872.

[262] J. Park, S. Samarakoon, M. Bennis, and M. Debbahu, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.

[263] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.

[264] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[265] Z. Li, Z. Yang, and S. Xie, "Computing resource trading for edge-cloud-assisted Internet of things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3661–3669, Jun. 2019.

[266] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Energy-aware QoE and backhaul traffic optimization in green edge adaptive mobile video streaming," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 3, pp. 828–839, Sep. 2019.

[267] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar./Apr. 2018.

[268] F. Qian, B. Han, J. Pair, and V. Gopalakrishnan, "Toward practical volumetric video streaming on commodity smartphones," in *Proc. 20th Int. Workshop Mobile Comput. Syst. Appl.*, 2019, pp. 135–140.

[269] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 116–122, Jun. 2018.

[270] M. S. Hossain, G. Muhammad, W. Abdul, B. Song, and B. B. Gupta, "Cloud-assisted secure video transmission and sharing framework for smart cities," *Future Gener. Comput. Syst.*, vol. 83, pp. 596–606, Jun. 2018.