Multimedia Tools and Applications (2021) 80:16007–16026
https://doi.org/10.1007/s11042-020-10393-8

Cross-layer DASH-based multipath video streaming over LTE and 802.11ac networks

Zhenjie Deng[1] · Yanwei Liu[2] · Jinxia Liu[3] · Antonios Argyriou[4]

Received: 11 September 2019 / Revised: 28 August 2020 / Accepted: 22 December 2020 /
Published online: 7 February 2021

Abstract

Dynamic Adaptive Streaming over Http (DASH) -based video streaming applications are becoming increasingly prevalent over the mobile Internet. Many efforts have been made to optimize their performances. Multipath video streaming that simultaneously utilizes multiple wireless networks for video content delivery is a common method. Another effective approach is the cross-layer video streaming optimization that jointly takes the parameters at different protocol layers into account. However, multipath streaming schemes mainly focus on how to efficiently utilize multiple wireless networks and the collaboration of parameters at different layers in each network is neglected. Likewise, the cross-layer schemes normally optimize the parameters at different layers in purely one network without fully utilizing the aggregated bandwidths of multiple available wireless networks. Therefore, both of them are sub-optimal and might suffer from degrading performance. In this paper, we propose a joint Cross-layer DASH-based multipath video streaming scheme that takes advantage of bandwidth aggregation of multiple wireless networks and further improves the performance by optimizing the different layers' parameters in each network with a cross-layer manner. In the proposed scheme, the LTE and 802.11ac networks are adopted. The bitrate of DASH-based video chunk at application layer, the rate allocation among networks and the Modulation and Coding Scheme (MCS) at physical layers in LTE and 802.11ac downlink are jointly optimized. We also compare our proposed scheme to state-of-the-art schemes using trace-driven experiments. Experimental results show that our proposed scheme outperforms state-of-the-art schemes in terms of PSNR, normalized QoE, and balance between video bitrate and rebuffering penalty.

Keywords DASH · Multipath · LTE · 802.11ac · Video streaming · MCS

✉ Zhenjie Deng
47-047@gduf.edu.cn

Yanwei Liu
liuyanwei@iie.ac.cn

[1] Guangdong University of Finance, Guangdong, China

[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

[3] Zhejiang Wanli University, Ningbo, China

[4] University of Thessaly, Volos, Greece

1 Introduction

Nowadays, video applications like Zoom, Youtube and Netflix are increasingly prevalent over the mobile Internet. Meanwhile, it is predicted by Cisco Visual Networking Index that mobile video traffic will upscale 9-fold from 2017 to 2022, and accounting for nearly four-fifths of total mobile data traffic by the end of the forecast period [8]. The major motivations behind this phenomenon are the proliferation of powerful mobile devices such as iPhone$^{\text{TM}}$-based and Android$^{\text{TM}}$-based smartphones, and the explosive demands for high quality video streaming from them. By and large, these mobile video streaming services are with high throughput and low latency requirements. Therefore, efforts should be made to optimize the delivery of mobile video streaming applications.

Among different video streaming standards and technologies, DASH is one of the dominant video streaming technologies over the mobile Internet [29]. Various proprietary proposals of DASH are developed, such as Apple HTTP Live Streaming, Microsoft Smooth Streaming and Adobe HTTP Dynamic Streaming [24]. The basic idea of DASH is that a video sequence is partitioned into multiple segments/chunks with constant playback length and replicas of each segment are stored in different sites in a Content Delivery Network (CDN) with different resolutions and qualities. DASH aims to adapt and optimize video streaming over time to offer the best possible video quality to the end user, by considering device capabilities, network conditions and content characteristics.

However, it is still challenging for video streaming over wireless cellular networks with guaranteed Quality of Experience (QoE) due to the limited capacity of the cellular network and the massive growth in mobile video traffic. A straightforward way to sustain the explosive growth of video traffic in the mobile network is to upgrade current cellular network to next generation advanced networks such as LTE-Advanced and 5G networks. Nevertheless, by simply increasing the capacity of cellular network might not always be economical [1]. Therefore, this approach requires continuous exploration of novel solutions for video streaming optimization in order to deliver an enhanced QoE for a wide range of mobile video applications.

With the development of techniques for simultaneous utilization of multiple network interfaces at the mobile devices, higher quality videos can be supported by using multiple wireless access networks simultaneously [29]. For example, in a place overlapped with both 802.11ac and LTE networks, a possible way to further enhance the video streaming performance is to download video chunks via LTE and 802.11ac interfaces simultaneously. Therefore, we propose to combine the DASH technique with multipath video streaming by delivering a video as a sequence of small, independent segments encoded in different bitrates and allowing a single video segment to be transported over various wireless links for bandwidth aggregation. To achieve this, the HTTP's *range retrieval requests* technique is adopted to enable a video segment to be logically partitioned and to be downloaded through various wireless network interfaces separately [3, 12, 29].

In addition to the adaptive bitrate at the application layer, the parameters at the physical layer such as MCS in both LTE and 802.11ac networks can be utilized to further optimize the DASH-based video streaming. Thus, a DASH-based cross-layer video optimization scheme is proposed in this paper to improve the perceptual video quality for end-to-end video streaming over multiple wireless access networks. In fact, the tuning of MCS at the physical layer in both 802.11ac and LTE networks, the video bitrate switching at the application layer are jointly performed by the cross-layer optimization controller (see

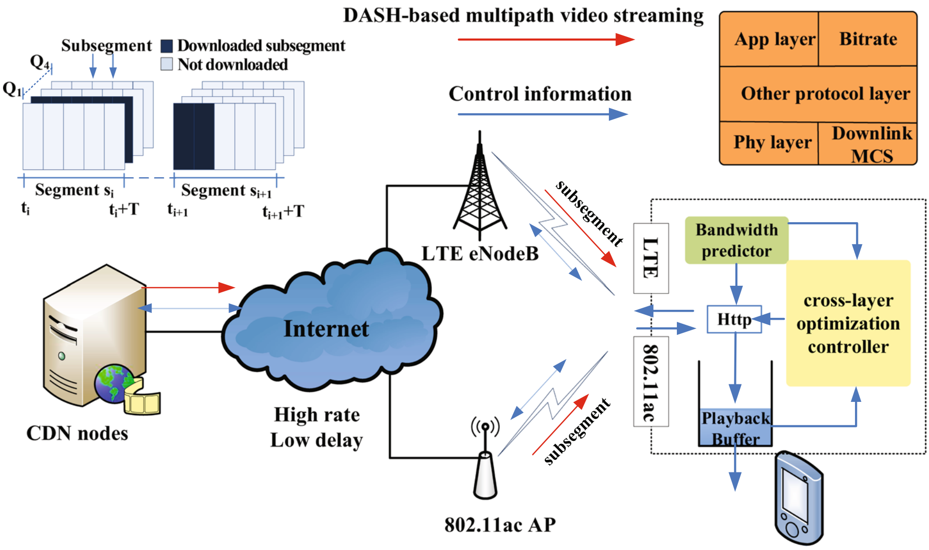

Fig. 1 The proposed DASH-based cross-layer optimization framework over multiple wireless networks

Fig. 1) according to the feedback information such as the Signal-to-Interference plus Noise Ratio (SINR) and the buffer occupancy rate. The major contributions of this paper can be summarized as follows:

- A DASH-based cross-layer optimization scheme is proposed for multipath video streaming over LTE and 802.11ac wireless networks. The MCS mode at the downlink physical layer and the video segment bitrate at the application layer are jointly adapted to enhance the video streaming performance.
- The playback buffer occupancy rate is also considered for bitrate selection and rate allocation between the LTE and 802.11ac networks. A logarithmic quality function is proposed to model the perceived QoE of each requested segment. Then we formulate this DASH-based cross-layer multipath video streaming problem as a nonlinear optimization problem with mixed discrete-continuous constraints and try to find the optimal bitrate, MCS and rate allocation values to maximize the nonlinear and non-differentiable objective function for each segment.
- To reduce the complexity, we propose an efficient online heuristic algorithm to find the sub-optimal solution to maximize the expected quality of the requested video segment and further evaluate its performance through a trace-driven simulation.

The rest of this paper is organized as follows. In Section 2, we discuss the related works concerning DASH-based video streaming in 802.11ac and LTE downlink networks. Section 3 describes the proposed DASH-based cross-layer multipath video streaming optimization framework, the tuning of parameters at the physical layer of LTE and 802.11ac networks, followed by the formulation of the optimization problem and the corresponding solution. In Section 4, we evaluate the performance of the proposed algorithm by trace-driven simulations, followed by the concluding remarks in Section 5.

2 Related work

To improve the quality of wireless video streaming from a cross-layer perspective, a variety of optimization schemes have been proposed. Zhao et al. [33] proposed a Structural SIMilarity index (SSIM)-based cross-layer optimized video streaming over LTE downlink wireless network. The MCS mode at the physical layer is selected to improve the perceptual video quality by jointly taking the characteristics of the video slice into account. In [2], Argyriou et al. investigated the performance of video streaming in heterogeneous cellular networks when the time-domain resource partitioning mechanism is employed. The perceived video quality for the subscribers is maximized by jointly optimizing the selected video quality transmitted to a user, the rate allocated to each specific user at the application layer, and the time-domain resource partitioning at the physical layer. In IEEE 802.11ac wireless local area networks, Chang et al. [6] proposed a cross-layer designed quality adaptive strategy to maximize the perceived H.264/AVC video streaming quality. A multi-polling controlled access (MPCA) scheme at the MAC layer and the video frame types at the application layer are jointly considered to guarantee the latency for the critical video frames and reduce transmission overhead. However, the above literatures [2, 6, 33] attempt to improve the video streaming performance by cross-layer method in one wireless network without taking the advantage of aggregated bandwidth from multipath video streaming.

At the application layer, HTTP-based adaptive video streaming (standardized as DASH [24]) is being widely adopted as a form of Internet video delivery. In [24], the standards and design principles of DASH specifications are presented and the implementation examples are also provided. In DASH, the adaptive bitrate (ABR) algorithm in the client is critical to ensure a desirable QoE and various ABR algorithms have been proposed. Previous ABR algorithms can be typically grouped into three classes: rate-based, buffer-based and reinforcement learning based methods. Rate-based algorithms [18, 25] usually request video segments at the highest bitrate that networks are predicted to support. However, these sort of methods first estimate the available bitrate by observing the past segment downloads which are often hindered by the biased throughput prediction on top of HTTP. In contrast, buffer-based methods merely keep track of the client's playback buffer occupancy while selecting the bitrates for later video segments. These methods strive to keep the buffer occupancy above a pre-configured threshold which balances video quality and rebuffering events. The most advanced buffer-based methods, both Buffer-Based Approach (BBA) [15] and Bitrate Adaptation for Online Video (BOLA) [23], are optimizing for a specified video quality metric only based on the observed buffer occupancy. Yin et al. [31] proposed a Model Predictive Control (MPC) algorithm which combines the rate-based and buffer-based techniques to select proper bitrates that expected to maximize the QoE over several future video segments. Nevertheless, MPC still suffers from inaccurate throughput estimation which is critical for its performance. The most recent reinforcement learning based approach, *Pensieve* [20], trained a neural network model to learn a precise ABR algorithm, and select bitrates automatically for a horizon of serval future segments. The *Pensieve* in the client learns the control policy for video bitrate adaptation purely through experience, without utilizing any specific assumptions or pre-configured models about the environment. To summarize, the above papers utilize adaptive bitrate algorithms to make video quality decision based on the predicted bandwidth or the buffer state of one wireless link, which also can further be optimized by jointly considering different parameters at different protocol layers or using multipath video streaming for bandwidth aggregation.

Leveraging both LTE and Wi-Fi links simultaneously can enhance the performance of video streaming services and therefore numerous DASH-based multipath video streaming schemes have been studied. In [19], the authors proposed a video segment request policy called *REQUEST* for DASH-based video streaming in a smartphone utilizing both Wi-Fi and LTE interfaces. *REQUEST* enables better video quality, fewer rebuffering events than other existing schemes under given budgets of LTE data usage and battery energy. In a multiuser scenario, Ho et al. [14] presented a game-theoretic scalable offloading framework that enabled seamless video streaming over LTE and Wi-Fi networks concurrently. In this framework, fountain encoding together with the progressive second price auction mechanism are employed to improve the video streaming performance among multiple smartphones. At the transport layer, the Multipath TCP (MPTCP) and the Multipath QUIC protocol [28] are designed to offer significant benefits to DASH-based multihomed video streaming. However, the congestion control algorithms in the above original multipath transport protocol are not suitable for multipath video streaming. James et al. [16] discussed that whether MPTCP is always beneficial for video streaming over DASH. They found that without sufficient bandwidth on the secondary path, the video streaming over MPTCP would suffer from degraded performance. Further, Han et al. [13] proposed a multipath framework called *MP-DASH* for video streaming over multiple network interfaces. *MP-DASH* strategically schedules video segments to satisfy user preferences. In order to provide a general framework, Chen et al. [7] proposed a DASH-based video streaming solution in the client-side, called *MSPlayer*, that exploiting multiple CDN nodes and network interfaces. *MSPlayer* provided the aggregating bandwidth for high-definition video streaming and reduced startup latency. However, *MSPlayer* does not assume multipath video streaming over MPTCP in which multiple transport links considered as one logical link to the application layer. In addition, *MSPlayer* doesn't provide strategy to select the wireless link. To address this, Elgabli et al. [11] proposed a preference-aware mulipath video streaming algorithm over HTTP using MPTCP. Howover, these MPTCP-based mulitpath video streaming strategies cannot be deployed without modifying the original congestion control algorithms. Therefore, MPQUIC protocol that using the UDP protocol in the transport layer is more suitable multipath video streaming. As a baseline of our scheme, Viernickel et al. [28] proposed Multipath-enabled QUIC (MPQUIC) solution to leverage multiple network interfaces to provide bandwidth aggregation. In this paper, We further improve the performance of multipath video streaming by adjust the MCS mode at the physical layer in a cross-layer method.

In summary, most existing video streaming solutions either purely rely on one network interface, or leverage multiple network interfaces without cross-layer optimization. Moreover, some researchers mainly make effort to find an optimal ABR algorithm by tuning the policy agent in the client to cater to the new environment. Motivated by the above analyses, we attempt to take advantage of the aggregated bandwidth from LTE and 802.11ac network interfaces, and exploit the cross-layer scheme to further improve the performance of DASH-based multipath video streaming. In the next section, we will describe the proposed DASH-based cross-layer optimization framework and the formulation of the optimizing problem over LTE and 802.11ac networks.

3 DASH-based multipath cross-layer optimization

3.1 DASH-based cross-layer optimization framework

Figure 1 shows the proposed DASH-based cross-layer multipath video streaming optimization framework. In this framework, the multi-interfaced (LTE and 802.11ac) client sequentially requests video segments stored in different CDN nodes via DASH technique over LTE and 802.11ac wireless network interfaces simultaneously. In the CDN side, the video sequence is partitioned into multiple independent segments, and each segment is with multiple replicas encoded with various bitrate values [7]. To fully take advantage of the aggregating bandwidth, each segment is logically divided into multiple subsegments, which can be requested through multiple wireless interfaces via *HTTP's range retrieval requests* [12, 29]. In such a scenario, two crucial issues should be considered in the client to ensure a good video streaming performance: how to select the bitrate for the new requested segment and how to slice each segment into two subsegments that delivered through LTE and 802.llac networks respectively.

To achieve this, the segment bitrate and the rate allocation at the application layer, the MCS mode at the downlink physical are jointly adjusted by the cross-layer optimization controller embedded in the client-side. When requesting a new segment, the link adaptation including the adjustment of MCS mode should be performed to adapt to the time-varying wireless channel states. Accordingly, the segment bitrate adjustments comprised of bitrate selection and rate allocation among separate links are dynamically tuned to match the integrated channel goodput that the selected MCS can support. In addition, the buffer occupancy in the client is also considered by the controller to avoid the rebuffering events.

The wireless channel is usually accompanied by time-varying characteristics and frequency-selective fading. To accommodate this, the Adaptive Modulation and Coding (AMC) is utilized to select the most suitable MCS mode based on the estimated channel state and Bit Error Rate (BER) /Block Error Rate (BLER). In practice, the MCS mode for a specific User Equipment (UE) is determined by the eNodeB/AP with the help of periodical feedback of Channel Quality Indicator (CQI) from the UE, which is represented by the Signal-to-Interference-plus-Noise-Ratio (SINR). For example, the MCS is selected to maintain the BLER of each resource block smaller than 10 percent for the LTE downlink channel adaptation [2, 33]. However, in our paper, the MCS mode is selected by considering both the SINR and the effect of its achieving goodput on the perceived video quality. In other words, the new segment bitrate value at the application layer should be selected up to the integrated bandwidth that LTE and 802.11ac downlink networks can support. Further, the rate allocation that determines the subsegment size transferred by the corresponding access networks is tuned to the selected MCS mode.

3.2 Video quality model for DASH

Two overarching goals have to be balanced in DASH-based video streaming applicatons. On one hand, they attempt to maximize the video quality of each video segment by selecting the highest video rate that networks can support, and maintaining a smooth video playback. On the other hand, they try to avoid rebuffering events that result in halt of video playback when the client's received buffer goes empty [6, 15, 20, 23, 31]. In this paper, the video is modelled as a sequence of consecutive video segments, $\mathcal{V} = \{1, 2, \cdots, K\}$, each of which contains T seconds of video and encoded with different bitrates. The player can choose to request a new segment with bitrate $r_i \in \mathcal{R}$, $i \in \mathcal{V}$, where $\mathcal{R}$ is the set of all available bitrate values. These information characterizing various representations of the media components (bitrates, resolutions, codecs, etc.) is contained in the media presentation description (MPD) file, which will be requested by the client during the initialization phase [24].

By neglecting the impact of rebuffering events and the quality variations between two consecutive segments, we denote $q(\cdot) : \mathcal{R} \rightarrow \mathbb{R}_+$ by the function which maps the selected video rate r_i of segment i to the perceptual video quality. According to [28], the perceptual video quality is increased with video bitrates. The slope is quite steep in the low bitrate region, but it gradually slows down at high bitrate values. The logarithmic function matches this characteristic well and is utilized to represent the video quality $q(\cdot)$ in this paper. Therefore, the perceptual video quality is expressed as

$$q(r_i) = \log\ (1 + \alpha \cdot r_i), \quad r_i \in \mathcal{R}, i \in \mathcal{V} \tag{1}$$

where α is a fitting parameter for a specific video codec and video sequence. It can be estimated from three or more trial encodings using nonlinear regression techniques.

To avoid the rebuffering events that strongly impair the user's experience, the current requesting segment has to arrive at the client before the playback buffer goes empty. Let t_b be the *buffer occupancy* at time t that starts to request segment i, i.e., the play time of the downloaded yet unviewed segment remained in buffer. The value of t_b can be obtained via periodical feedback by the client to the optimization controller. We also denote by C_s the average total goodput provided by all the access networks from moment t_i to $t_i + T$. Note that if $T \cdot r_i / C_s \geq t_b$, the buffer goes empty while the client is still downloading segment i, resulting in rebuffering events [15, 20, 29]. We define a tradeoff function to balance the impairment of rebuffering and the video playback quality. A tradeoff coefficient λ is introduced to weight the impairment of the rebuffering events. This modified perceived video quality function can be represented as

$$Q(r_i) = q(r_i) - \lambda \cdot \mathrm{I}\ (T \cdot r_i / C_s - t_b) \tag{2}$$

where $\mathrm{I}\ (\cdot)$ is the step function that $\mathrm{I}\ (\cdot) = 1$, if $T \cdot r_i / C_s \geq t_b$, otherwise, $\mathrm{I}\ (\cdot) = 0$.

Since each segment is logically divided into two subsegments, each of which will be requested over the LTE and 802.11ac downlink simultaneously via the *HTTP's range retrieval requests* technique [12, 29]. The rebuffering event occurs if one of the subsegments cannot arrive at the client before the playback buffer runs out. Let $r_{i,1}$ and $r_{i,2}$ be the bitrates allocated to LTE and 802.11ac wireless networks respectively. Their sum equals to the selected bitrate r_i of segment i. That is $r_i = r_{i,1} + r_{i,2}$. The average downlink goodput provided by the LTE and 802.11ac wireless networks while downloading the subsegments are denoted by $C_{i,1}$ and $C_{i,2}$ respectively. In this case, the rebuffering event emerges if $\max(T \cdot r_{i,1}/C_{i,1}, T \cdot r_{i,2}/C_{i,2}) \geq t_b$. Thus, the ultimate quality function for segment i can be defined as

$$Q(r_i, r_{i,1}, r_{i,2}) = q(r_i) - \lambda \cdot \mathrm{I}\ (\max(\frac{T \cdot r_{i,1}}{C_{i,1}}, \frac{T \cdot r_{i,2}}{C_{i,2}}) - t_b) \tag{3}$$

3.3 The goodput estimation of LTE downlink

In the LTE downlink, the achieved goodput depends on the wireless channel condition, the selected MCS mode and the resource allocation algorithm. To estimate the effective average goodput $C_{i,1}$ while downloading the corresponding subsegment through LTE downlink, the mutual information effective SNR mapping (MIESM) is utilized to measure the LTE downlink channel quality in this paper. For the selected MCS mode $m_1 \in \mathcal{M}_1$, where $\mathcal{M}_1$

is the candidate MCS mode set in the first column of Table 1, the effective SNR mapping $\gamma_{mieff}(m_1)$ based on the mutual information can be calculated as [17]

$$\gamma_{mieff}(m_1) = \tau(m_1)\left[J^{-1}\left(\frac{1}{S_n}\sum_{k=1}^{S_n} J\left(\sqrt{\frac{\gamma_k}{\tau(m_1)}}\right)\right)\right]^2 \tag{4}$$

where S_n is the number of allocated subcarriers for subsegment i, $\tau(m_1)$ is the calibration factor for MCS mode m_1 listed in Table 1, and γ_k is the SINR at the k^{th} subcarrier. The definition of functions $J(\cdot)$ and $J^{-1}(\cdot)$ are defined as (5) and (6). For more details, please refer to the references [10, 17, 33].

$$J(x) \approx \begin{cases} -0.04210610x^3 + 0.209252x^2 - 0.00640081x, & \text{0 ¡ x ¡ 1.6363} \\ 1 - \exp(0.00181491x^3 - 0.142675x^2 - 0.08220540x + 0.0548608), & \text{x} \geq 1.6363 \end{cases} \tag{5}$$

$$J^{-1}(y) \approx \begin{cases} 1.09542y^2 + 0.214217y + 2.33727\sqrt{y}, & \text{0 ¡ y ¡ 0.3646} \\ \text{- 0.706692log(- 0.386013(y - 1)) + 1.75017y}, & 0.3646 \leq \text{y} \leq 1 \end{cases} \tag{6}$$

Based on the MIESM $\gamma_{mieff}(m_1)$ defined in (4), the Block Error Rate (BLER) $BLER(\gamma_{mieff}(m_1))$ for the RB with MCS mode m_1 can be precisely predicted as

$$BLER(\gamma_{mieff}(m_1)) = \frac{1}{2}erfc(\frac{\gamma_{mieff}(m_1) - b(m_1)}{\sqrt{2}c(m_1)}) \tag{7}$$

where $erfc(\cdot)$ is the complementary error function, $b(m_1)$ and $c(m_1)$ listed in Table I are the "transition center" and "transition width" respectively, each of which can be obtained by fitting $J^{-1}(\cdot)$ to the exact BLER in a specific communication system. In this paper, a MIMO 2X1 AWGN LTE downlink channel is simulated using a generic LTE system-level simulator in [26].

Table 1 The candidate LTE downlink MCS modes

MCS index(m_1)	Modulation order	Rate(bits /symbol $r(m_1)$)	$\tau(m_1)$	$b(m_1)$	$c(m_1)$
1	QPSK	0.1523	3.07	-7.758	0.6003
2	QPSK	0.2344	4.41	-5.724	0.5182
3	QPSK	0.3770	0.60	-3.652	0.4032
4	QPSK	0.6010	1.16	-1.593	0.3588
5	QPSK	0.8770	1.06	0.3501	0.2910
6	QPSK	1.1758	1.06	2.348	0.2563
7	16QAM	1.4766	0.87	4.297	0.2563
8	16QAM	1.9141	1.01	6.214	0.2293
9	16QAM	2.4063	1.04	8.242	0.2253
10	64QAM	2.7305	1.03	10.13	0.2248
11	64QAM	3.3223	1.11	12.06	0.2028
12	64QAM	3.9023	1.01	13.89	0.1962
13	64QAM	4.5234	1.07	15.72	0.1958
14	64QAM	5.1152	1.00	17.50	0.2134
15	64QAM	5.5547	1.05	19.59	0.2592

Due to the truncated ARQ mechanism implemented in the data link layer, resource blocks that are received in error during the original transmission might be retransmitted, up to a maximum of N_r times. For notational simplicity, let us define $\epsilon(m_1) \overset{\text{def}}{=} BLER(\gamma_{mieff}(m_1))$, and the average number of transmissions per resource block can be derived as

$$\begin{aligned}\overline{N}(\epsilon(m_1), N_r) &= \sum_{i=1}^{N_r} i \cdot (1-\epsilon(m_1)) \cdot \epsilon(m_1)^{i-1} \\ &= 1 + \epsilon(m_1) + \epsilon(m_1)^2 + \cdots + \epsilon(m_1)^{N_r} \\ &= \frac{1-\epsilon(m_1)^{N_r}}{1-\epsilon(m_1)} \end{aligned} \tag{8}$$

To evaluate the achieved channel goodput, the number of information bits carried by each transmitted symbol is calculated as $r(m_1) = R_c \cdot \log_2(M_{m_1})$ and listed in Table 1, where R_c is the FEC code rate and M_{m_1} refers to a M_{m_1}-QAM constellation for MCS mode m_1.

It has been known that the available spectrum resource is divided into some individual resource blocks based on the frequency and time domains in LTE downlink physical layer. Each RB occupies the duration of one slot (0.5ms) and contains 7 OFDM symbols with normal cyclic prefix in the time domain and 12 subcarriers (180KHz) in the frequency domain. However, three downlink control channels are defined in the LTE downlink in order to support the data transmission, which are Physical Control Format Indicator Channel (PCFICH), Physical HARQ Indicator Channel (PHICH), Physical Downlink Control Channel (PDCCH). In the normal configuration, these channels occupy the the first three OFDM symbols in each sub-frames (1ms) in the time domain and the whole bandwidth in the frequency domain, described by the grey square blocks in Fig. 2. We can see in Fig. 2 that there are eight resource elements reserved for reference signals in each resource block [22]. Therefore, the available data bits carried by two adjacent RBs in one sub-frame, as a function of MCS mode m_1, can be expressed as $\xi(m_1, N_r) = N_{rb} \cdot r(m_1)$, where $N_{rb} = 120$ denotes the number of resource element allocated for the data transmission in two adjacent RBs.

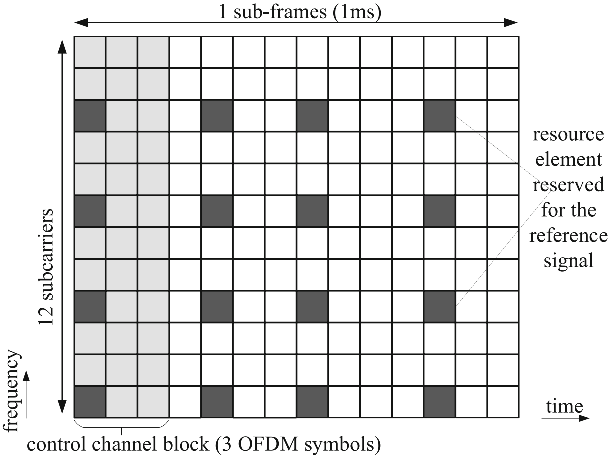
1 sub-frames (1ms)
12 subcarriers
frequency
resource element reserved for the reference signal
time
control channel block (3 OFDM symbols)

Fig. 2 The frequency and time domains in the LTE downlink

In each Transmission Time Interval (TTI), the Proportional Fair Scheduling (PFS) algorithm [4] is used for resource block scheduling among multiple users in one single cell. Suppose the total RB numbers allocated for the delivery of subsegment in LTE downlink equals to B_n and all the RBs adopt the same MCS mode. When the truncated ARQ is adopted, each resource block is averagely transmitted $\overline{N}(\epsilon(m_1), N_r)$ times. Therefore, the achieved goodput can be computed as

$$C_{i,1}(m_1) = \frac{\xi(m_1, N_r) \cdot B_n}{\overline{N}(\epsilon(m_1), N_r)} \tag{9}$$

3.4 The goodput estimation of 802.11ac network

In the 802.11ac downlink physical layer, OFDM is selected as the modulation scheme and ten MCS modes with different modulation schemes and coding rates are provided for link adaptation. Specifically, BPSK, QPSK, 16-QAM, 64-QAM and 256-QAM are the supported modulation schemes listed in Table 2. In the MAC layer, to share the wireless channel between multiple compatible stations, the contention-based Distributed Coordination Function (DCF) that uses the algorithm of Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) is implemented as a mandatory medium access control (MAC) mechanism. In CSMA/CA, each successful frame transmission duration of DCF consists of a backoff delay $\bar{T}_b$, the data transmission time $T_{data}(l, m_2)$, a Short InterFrame Space (SIFS) time $T_{\text{SIFS}} = 16\mu s$, the ACK transmission time $T_{ack}(m_2)$ and a Distributed InterFrame Space (DIFS) time $T_{\text{DIFS}} = 34\mu s$ [9]. Suppose that a frame with l bits data payload is to be transmitted using MCS mode $m_2 \in \mathcal{M}_2$, where $\mathcal{M}_2$ is the candidate MCS mode set in the first column of Table 2. According to [30], the data transmission duration can be calculated as

$$T_{data}(l, m_2) = 20\mu s + \lceil \frac{30.75 + l}{r(m_2)} \rceil \cdot 4\mu s \tag{10}$$

where $r(m_2)$ can be computed by the code rate given in Table 2 and is the bits-per-symbol information for MCS mode m_2. For simplicity, the same MCS mode is supposed to used for the ACK frame transmission. The duration for an ACK frame can be expressed as follows [30],

$$T_{ack}(m_2) = 20\mu s + \lceil \frac{16.75}{r(m_2)} \rceil \cdot 4\mu s \tag{11}$$

Table 2 Data rates (Mbps)-ten MCS modes, 1 spatial stream, normal guard interval

MCS Index(m_2)	Modulation	Code Rate	20Mhz	40Mhz	80Mhz
0	BPSK	1/2	6.5	13.5	29.3
1	QPSK	1/2	13	27	58.5
2	QPSK	3/4	19.5	40.5	87.8
3	16-QAM	1/2	26	54	117
4	16-QAM	3/4	39	81	175.5
5	64-QAM	2/3	52	108	234
6	64-QAM	3/4	58.5	121.5	263.3
7	64-QAM	5/6	65	135	292.5
8	256-QAM	3/4	78	162	351
9	256-QAM	5/6	-	180	390

In the backoff period, a random integer is assigned to the station according to a uniform distribution over the interval [0, CW], where CW is the content window size and its initial value is CW_{min}. Based on the formulation in [21], the average backoff time is given by

$$\bar{T}_b = \frac{CW_{min} \cdot T_{\text{slot}}}{2} \tag{12}$$

where T_{slot} is the slot time in 802.11ac and is equal to 9μs.

A frame transmission is considered successful only upon receiving the corresponding ACK frame correctly. Therefore, the probability of a successful frame transmission with wireless channel state γ_2 and MCS mode m_2 can be calculated by

$$P_s(l, \gamma_2, m_2) = [1 - P_{data}(l, \gamma_2, m_2)][1 - P_{ack}(\gamma_2, m_2)] \tag{13}$$

where $P_{data}(l, \gamma_2, m_2)$ and $P_{ack}(\gamma_2, m_2)$ are the data error probability and the ack error probability, respectively, and their values are varied under different wireless channel model and estimated over the AWGN channel in this paper. Since the data frame is normally much longer than the ACK frame, the probability for the ACK frame to be lost is much smaller than the data frame. Thus, we have the following approximation

$$P_s(l, \gamma_2, m_2) \approx 1 - P_{data}(l, \gamma_2, m_2) \tag{14}$$

An upper bound is given on the packet error probability, under the assumption that hard-decision Viterbi decoding with independent errors and binary convolutional coding are used at the channel input. The data packet error probability with l octets using MCS mode m_2 is bounded by

$$P_{data}(l, \gamma_2, m_2) \leq 1 - (1 - P_u(\gamma_2, m_2))^{8l} \tag{15}$$

where the union bound $P_u(m_2)$ is the first-event error probability given by

$$P_u(\gamma_2, m_2) = \sum_{d=d_{free}(m_2)}^{\infty} a_d \cdot P_d(\gamma_2) \tag{16}$$

where $d_{free}(m_2)$ is the free distance for the convolutional code in MCS mode m_2, a_d is total number of error of weight d, and $P_d(\gamma_2)$ is the probability of an incorrect path at distance d from the correct path being chosen by the Viterbi decoder and is given as follows,

$$P_d(\gamma_2) = \begin{cases} \sum\limits_{i=(d+1)/2}^{d} \binom{d}{i} \cdot \rho^i \cdot (1-\rho)^{d-i}, \text{ if } d \text{ is odd,} \\ \frac{1}{2} \cdot \binom{d}{d/2} \cdot \rho^{d/2} \cdot (1-\rho)^{d/2} + \\ \sum\limits_{i=d/2+1}^{d} \binom{d}{i} \cdot \rho^i \cdot (1-\rho)^{d-i}, \text{ if } d \text{ is even} \end{cases} \tag{17}$$

Note that ρ is the bit-error-rate as a function of the symbol SNR γ_2 for the MCS mode m_2 and can be approximated by (18) [32].

$$\rho = \frac{\sqrt{m_2} - 1}{\sqrt{m_2}\log_2\sqrt{m_2}} erfc\left(\sqrt{\frac{3\log_2(m_2 \cdot \gamma_2)}{2(m_2 - 1)}}\right) \tag{18}$$

Based on the above analysis, the effective goodput $C_{i,2}(m_2)$ of IEEE 802.11ac network can be calculated by

$$C_{i,2}(m_2) = \frac{P_s(l, \gamma_2, m_2) \cdot l}{\bar{T}_b + T_{data}(l, m_2) + T_{ack}(m_2) + T_{\text{SIFS}} + T_{\text{DIFS}}} \tag{19}$$

3.5 Optimization formulation and solution

With the feedback of the effective goodput estimation and the *buffer occupancy*, the controller attempts to maximize the perceptual video quality for each segment without causing the rebuffering events. The decision variables include the requested segment bitrate r_i, the rate allocation $r_{i,1}$ and $r_{i,2}$ from the application layer and the MCS mode m_1 and m_2 at the physical layer of different networks. In addition, to offer a smooth playback, the quality variation between two consecutive segments should be smaller than a threshold μ preferred by the user. Therefore, the cross-layer optimization problem can be formulated as

$$\begin{aligned} \max_{\{r_i, r_{i,1}, r_{i,2}, m_1, m_2\}} & \left\{ q(r_i) - \lambda \cdot \mathrm{I}\left(\max\left(\frac{r_{i,1}}{C_{i,1}(m_1)}, \frac{r_{i,2}}{C_{i,2}(m_2)} \right) - t_b \right) \right\} \\ s.t. r_i &= r_{i,1} + r_{i,2} \\ |q(r_i) &- q(r_{i-1})| \leq \mu \\ & r_i \in \mathcal{R} \\ & m_1 \in \mathcal{M}_1 \\ & m_2 \in \mathcal{M}_2 \end{aligned} \tag{20}$$

where the effective goodput $C_{i,1}(m_1)$ and $C_{i,2}(m_2)$, as a function of MCS mode and SNR, can be calculated by (9) and (19), respectively. For more information about the parameters in this paper, please refer to Table 3.

It can be seen that (20) contains both discrete and continuous variables. For instance, m_1 and m_2 are discrete while $r_{i,1}$, $r_{i,2}$ are continuous. Furthermore, the objective function in (20) is nonlinear and non-differentiable. Therefore, the cross-layer optimizing problem of (20) is a typical nonlinear optimization problem with mixed constraints. These kinds of problems are NP-hard without polynomial time solution. To solve the cross-layer optimizing problem formulated in (20), we construct a heuristic algorithm to find the near-optimal decision variables $(r_i^*, r_{i,1}^*, r_{i,2}^*, m_1^*, m_2^*)$ to maximize the perceived video quality of segment i. That is $Q(r_i^*, r_{i,1}^*, r_{i,2}^*, m_1^*, m_2^*) \geq Q(r_i, r_{i,1}, r_{i,2}, m_1, m_2), \forall\, r_i, r_{i,1}, r_{i,2}, m_1, m_2$ subject to the constraints defined in (20). In the algorithm, we first pick up a candidate bitrate set $\mathcal{R}_{candidate}$ that satisfies $|q(r_i) - q(r_{i-1})| \leq \mu$ and then sort the elements in the candidate set by descent order. In other words, the quality variation caused by two consecutive segments are tolerable if the bitrate of segment i is one of the elements in $\mathcal{R}_{candidate}$.

Table 3 Summary of terminology

Term	Brief Description
$r_i, \mathcal{R}$	bitrate of segment i, video bitrate set
$m_i, \mathcal{M}_i$	selected MCS mode for segment i, MCS mode set
$q(\cdot), Q(\cdot), I(\cdot)$	video quality function, indicator function
$C_{i,1}, C_{i,2}$	average downlink goodput of given network for segment i
T, t_b	the time duration of each segment, the buffer occupancy
γ	the SINR
α, μ, λ	fitting parameters, all set to 1 in the experiment

Algorithm 1 The proposed heuristic algorithm.

```
Input: $\gamma, \mathcal{R}, \mathcal{M}_1, \mathcal{M}_2, \alpha, \lambda, \mu, t_b, q(r_{i-1})$
Output: $r_i^*, r_{i,1}^*, r_{i,2}^*, m_1^*, m_2^*$
Initialization: $\mathcal{R}_{candidate} \leftarrow \emptyset$
for $r_i \in \mathcal{R}$ do
    if $|q(r_i) - q(r_{i+1})| > \mu$ then
        don't request the segment with rate $r_i$
    else
        $\mathcal{R}_{candidate} \leftarrow \mathcal{R}_{candidate} \cup r_i$
        sort the elements in $\mathcal{R}_{candidate}$
    end
end
while $\mathcal{R}_{candidate} \neq \emptyset$ do
    $r_i^* = \arg\max_{r_i} r_i, r_i \in \mathcal{R}_{candidate}$
    while $\mathcal{M}_1 \neq \emptyset, \mathcal{M}_2 \neq \emptyset$ do
        $m_1^* = \arg\min_{m_1} m_1, m_1 \in \mathcal{M}_1$
        $m_2^* = \arg\min_{m_2} m_2, m_2 \in \mathcal{M}_2$
        estimate goodput $C_{i,1}(m_1^*), C_{i,2}(m_2^*)$ based SNR and MCS
        $r_{i,1}^* = \frac{C_{i,1}(m_1^*)}{C_{i,1}(m_1^*)+C_{i,2}(m_2^*)} \cdot r_i^*$
        $r_{i,2}^* = \frac{C_{i,2}(m_2^*)}{C_{i,1}(m_1^*)+C_{i,2}(m_2^*)} \cdot r_i^*$
        if $\max\left(\frac{r_{i,1}^*}{C_{i,1}(m_1^*)}, \frac{r_{i,2}^*}{C_{i,2}(m_2^*)}\right) \leq t_b$ then
            return $(r_i^*, r_{i,1}^*, r_{i,2}^*, m_1^*, m_2^*)$
        else
            $\mathcal{M}_1 \leftarrow \mathcal{M}_1 \setminus m_1^*$
            $\mathcal{M}_2 \leftarrow \mathcal{M}_2 \setminus m_2^*$
        end
    end
    $\mathcal{R}_{candidate} \leftarrow \mathcal{R}_{candidate} \setminus r_i^*$
end
```

Since the video quality function $q(\cdot)$ is increasing with the bitrate r, we aggressively request segment i with the highest bitrate values in the candidate set. That is $r_i^* = \arg\max_{r_i} r_i, r_i \in \mathcal{R}_{candidate}$. After selecting the optimal bitrate r_i^* at the application layer, we will determine the MCS mode at the physical layer for the corresponding network. The MCS mode with small constellation and powerful channel code can maintain reliability at poor channel condition. Therefore, we select the MCS mode with smallest constellation, and channel code and estimate the achievable goodput $C_{i,1}(m_1^*)$ and $C_{i,2}(m_2^*)$ based on the selected MCS mode and symbol SNR.

The rate allocation that determines the size of the subsegment is based on the goodput of each network. That is $r_{i,1}^* = \frac{C_{i,1}(m_1^*)}{C_{i,1}(m_1^*)+C_{i,2}(m_2^*)} \cdot r_i^*$, $r_{i,2}^* = \frac{C_{i,2}(m_2^*)}{C_{i,1}(m_1^*)+C_{i,2}(m_2^*)} \cdot r_i^*$. Then the optimal decision variables $(r_i^*, r_{i,1}^*, r_{i,2}^*, m_1^*, m_2^*)$ are obtained so far. However, such a decision variable set might lead to rebuffering event. We assume a relatively large λ indicates that the user is more concerned about rebuffering is used in this algorithm. So every subsegment has to arrive at the client before the received buffer run out. Note that if

$\max\left(\frac{r^*_{i,1}}{C_{i,1}(m^*_1)}, \frac{r^*_{i,2}}{C_{i,2}(m^*_2)}\right) \leq t_b$, the rebuffering event will not occur and the decision variables are verified. Otherwise, the effective goodput given the selected MCS mode cannot satisfy the quality level of video segment i with bitrate of r^*_i and the MCS mode with larger constellation size and more powerful channel code is selected as the optimal MCS mode. If there is no MCS mode can satisfy such a bitrate level of segment i, the algorithm will select segment i with a smaller bitrate level. The details of the proposed heuristic algorithm for cross-layer multi-path streaming is shown in Algorithm 1. To evaluate our proposed heuristic algorithm, we construct a off-line mapping table between the goodput C of network and the candidate MCS mode m. Based on this, our heuristic algorithm is with polynomial time complexity. Specifically, in the first phase of the algorithm, we attempt to determine the candidate bitrate set $R_{candidate}$ within the available video bitrate. The time complexity is $\mathcal{O}(R)$ and linear. In the second phase of the algorithm, we try to find the appropriate video bitrate allocated to different network and the MCS mode. By using the off-line mapping table aforementioned, we can obtain the goodput C in constant time. Therefore, the total complexity of our heuristic algorithm is $\mathcal{O}(R \cdot M_1 \cdot M_2)$ within polynomial time.

4 Evaluation

4.1 Experimental setup

At the application layer, the video sequences are encoded via the H.264/AVC reference software JM18.6 [27] by setting different quantization parameters (QP). In our setup, each video segment is encoded at bitrate values in {350, 700, 1200, 1800, 2800, 4500} kbps, corresponding to various resolutions in { 240p, 360p, 480p, 720p, 1080p, 1440p}. We can also see from (2) that the video quality is related to two factors: the video segment bitrates and the rebuffering events. Hence, besides the QP at the application layer that determines the video bitrates, we also slice each whole video sequence into 50 segments and had a total duration of 200 seconds, which each segment stands for approximately 4 seconds of playback. In the simulation, we assume that the video player at the client was configured to hold a buffer capacity with enough playback duration.

Additionally, the LTE and 802.11ac downlink wireless channel are simulated through MATLAB Software based on [26] and [9], respectively. Then we exploit these generated traces to evaluate the performance of the proposed algorithm. The main experimental parameters for both video coding and the wireless network environment are shown in Table 4. To evaluate the proposed scheme, we compared it with state-of-the-art schemes including the Multipath QUIC procotol (MPQUIC) scheme [28] and the SSIM-based Cross-layer optimization with Error-resilient RDO (SSIM-CL-w-ERDO) scheme [33]. In the SSIM-CL-w-ERDO scheme, we split each segment into two subsegments with equal size and each of them is optimized by the SSIM-CL-w-ERDO scheme in LTE and 802.11ac downlink, respectively. These three schemes are evaluated under different wireless channel conditions (Rayleigh distribution with average SINR $\overline{\gamma}$ at 4dB, 9dB, 14dB) [5].

4.2 Experimental results

In the proposed DASH-based cross-layer multi-path video streaming scheme, the video quality experienced by the end user is optimized by adaptively selecting the video bitrates, the MCS mode for each segment according to the wireless channel state of both LTE and

Table 4 Experimental parameters

Video coding profile	baseline
Video coding structure	IPPP
Video frame rate	30fps
Error concealment	Temporal replacement
video bitrates	{350, 700, 1200, 1800, 2800, 4500} kbps
video resolution	{240p, 360p, 480p, 720p, 1080p, 1440p}
Video sequence	{ Shield, Parkrun, ElephantsDream}
Segment lengths	4 seconds
Candidate MCS modes	Tables 1 and 2
Average SINR $\overline{\gamma}$	(4, 9 ,14)dB
SINR distribution	Rayleigh

802.11ac downlink. Firstly, we investigated whether our proposed scheme can obtain the anticipated results. The adaptive selection of the MCS modes and the bitrates for total 50 segments of the video sequence *ElephantsDream* at different channel conditions (average SINR $\bar{\gamma} = 4dB$, $\bar{\gamma} = 9dB$, $\bar{\gamma} = 14dB$ for both LTE and 802.11ac downlink channel) are shown in Fig. 3.

MCS mode Adaptation We can see from Fig. 3 that, at a good channel condition of $\bar{\gamma} = 14dB$, the MCS modes with large size constellations (large MCS mode indexes) in both LTE and 802.11ac downlink and large video segment bitrates are selected to improve the video perceptual quality. In the other hand, at a poor channel condition of $\bar{\gamma} = 4dB$, the MCS modes with small size constellations (small MCS mode indexes) in both two networks and small bitrate are selected to avoid the rebuffering events and guarantee the smoothness of video playback. Therefore, it illustrates that our proposed scheme can effectively adjust the MCS mode at the physical layer in both LTE and 802.11ac and bitrates of the video segment to improve the video streaming performance in a cross-layer manner.

Segment-level analysis To evaluate the streaming performance of the proposed scheme, the segment-level average PSNR values for 50 segments of the three video sequences *Parkrun*, *Shield* and *ElephantsDream* transmitted at the channel condition of $\bar{\gamma} = 9dB$ are shown in Fig. 4. From Fig. 4, it can be observed that our proposed scheme can achieve higher

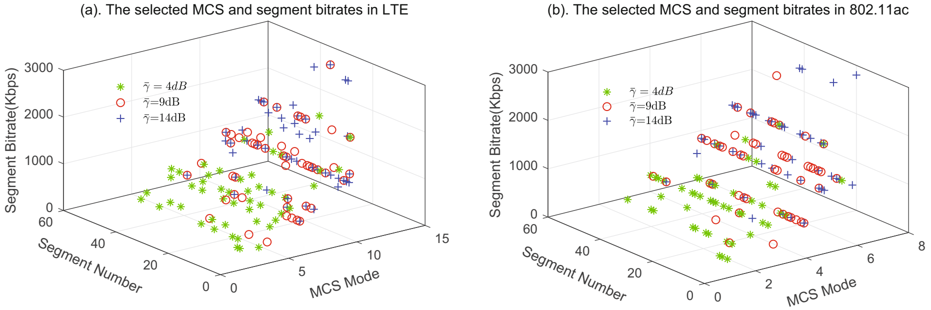

Fig. 3 The selected MCS modes and bitrates for total 50 segments of the video sequence *ElephantsDream* at different channel conditions of LTE and 802.11ac

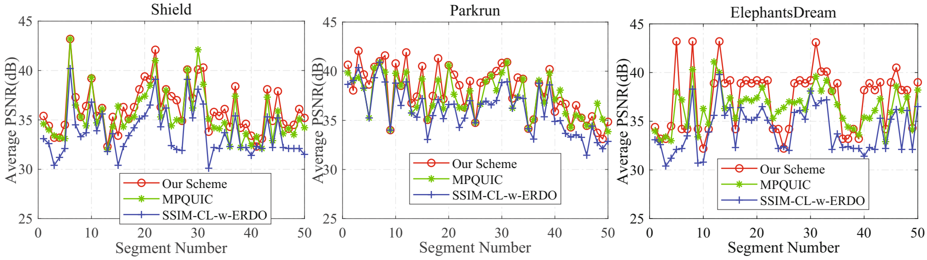

Fig. 4 Average PSNR curves with increasing segment numbers for the sequence **a** *Shield*, **b** *Parkrun* and **c** *ElephantsDream* at the condition of $\bar{\gamma} = 9dB$.

average PSNR values than other two baseline schemes (MPQUIC and SSIM-CL-w-ERDO) for most of the video segments. However, Fig. 4 additionally illustrates that our scheme does not overwhelm the other two baseline schemes on every segment level. This is because the balance between the bitrate utility and rebuffering penalty in our scheme. On the whole, the average PSNR of the total 50 segments of the *Shield* video sequences obtained by our proposed scheme is 1.16dB and 2.52dB higher than the two baseline schemes, respectively. For the video sequence of *Parkrun*, our proposed scheme outperformed the baseline schemes by 0.80dB and 1.91dB, respectively. For the video sequence of *ElephantsDream*, the average PSNR improvement is 1.28dB and 2.80dB, respectively.

Video sequence level analysis Figure 5 shows the average PSNR curves of the video sequences *Shield*, *Parkrun* and *ElephantsDream* at the wireless channel conditions of $\bar{\gamma} = 2dB, \bar{\gamma} = 4dB, \bar{\gamma} = 9dB, \bar{\gamma} = 14dB, \bar{\gamma} = 20dB$, respectively. It can be seen that our proposed scheme can achieve higher average PSNR values than other two baseline schemes in all wireless channel conditions. On average, for the video sequence of *Shield*, the average PSNR value achieved by our proposed scheme is approximately 1.55dB and 2.94dB than other baseline schemes. For the video sequence of *Parkrun*, the improvement is about 1.03dB and 2.78dB, respectively. For the video sequence of *ElephantsDream*, our scheme can overwhelm the two baseline schemes by 1.10dB and 2.31dB, respectively. Additionally, the performances of the average PSNR curves versus SINR show some differences under different channel conditions. It can also be observed from Fig. 5 that when SINR is with small value, in other words, when the channel condition is poor, our proposed scheme achieves a higher improvement of PSNR than that of average SINR with high value

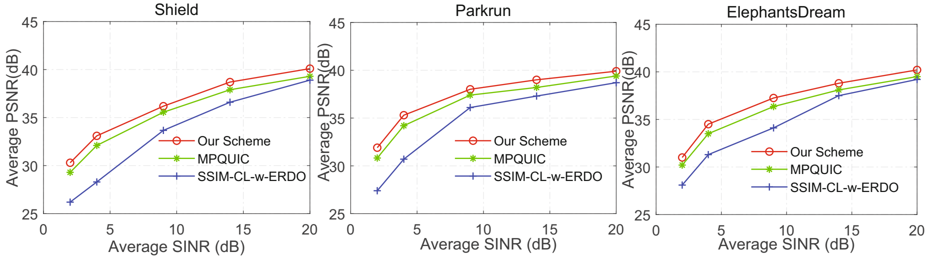

Fig. 5 Average PSNR curves of our scheme and two baseline schemes for the sequence **a** *Shield*, **b** *Parkrun* and **c** *ElephantsDream* at different channel conditions.

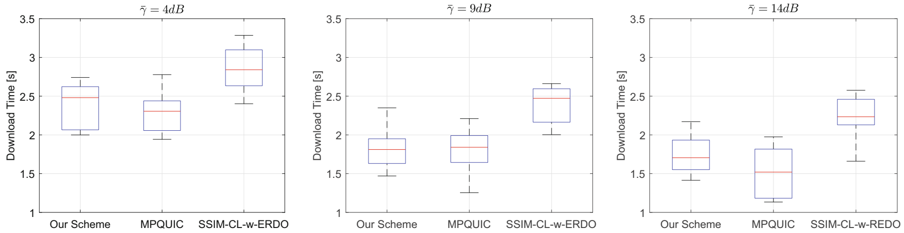

Fig. 6 Download time of each segment at different channel condition

(the wireless channel quality is good). For instance, while streaming the sequence of *ElephantsDream* at the condition of $\bar{\gamma} = 4dB$, the average PSNR achieved by our scheme is approximately 1.32dB and 3.21dB higher than the MPQUIC and SSIM-CL-w-ERDO, respectively. However, when the wireless channel quality is good, for example, at the condition of $\bar{\gamma} = 14dB$, the improvement is just 0.80dB and 1.30dB corresponding to the two baseline schemes, respectively. These could be due to the adaptive selected MCS mode at the physical layers in both LTE and 802.11ac that meeting the bitrate and rebuffering requirements for the streaming of each video segment.

Delay performance To evaluate the delay performance of the proposed scheme, the comparison of download time for each video segment at different channel condition between our scheme and two baseline scheme (MPQUIC and SSIM-CL-w-REDO) is showed in Fig. 6. We notice from Fig. 6 that when the channel condition become better, the video segment can be downloaded in less time for all the three schemes. Furthermore, though our scheme can achieve shorter download time performance than the SSIM-CL-w-REDO approach in different channel condition, we note that our scheme suffer from a little higher download time than the MPQUIC approach. MPQUIC can achieve better delay performance because it utilizes UDP protocol at the transport layer. Instead of UDP protocol used by MPQUIC, the TCP protocol used by our scheme or other MPTCP-based approach will introduce more delay by the acknowledgement mechanism.

QoE To better comprehend the QoE gains obtained by our scheme, we evaluate the performance on the individual terms in the QoE model that defined in (3). Explicitly, Fig. 7 shows the comparison between our scheme and two baseline schemes (MPQUIC and SSIM-CL-w-ERDO) in different channel conditions in terms of the playback bitrate utility from the first

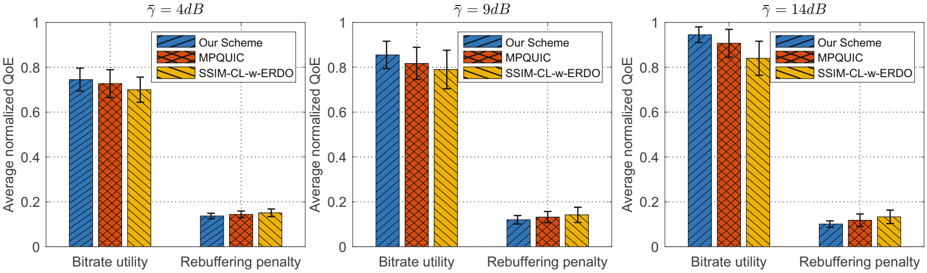

Fig. 7 Comparing our scheme with the baseline algorithms on the QoE metrics in terms of bitrate utility and rebuffering penalty

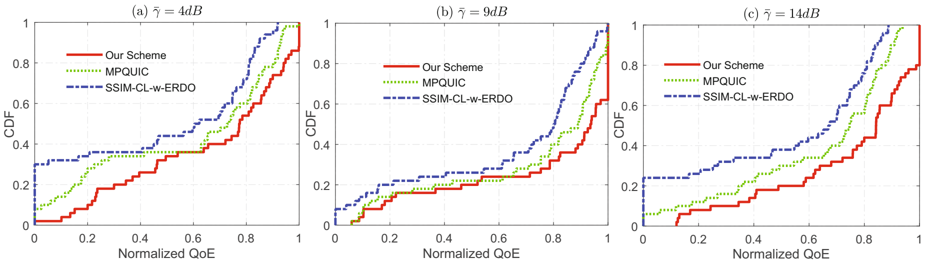

Fig. 8 The Cumulative Distribution Function (CDF) of the normalized QoE value across three different channel conditions

term in (3), and the penalty of rebuffering from the second term in (3). More precisely, the given QoE value can be calculated by subtracting the rebuffering penalty from the bitrate utility.

The performance gains of our scheme contribute to the aggregated bandwidth and adaptive selection of MCS mode to support higher video bitrate, and the ability to try to avoid rebuffering event from network's bandwidth fluctuations. As shown from Fig. 7, all the schemes obtain better bitrate utility values in the first term of (3) as the improvement of network channel condition. But we also see that the achieved bitrate utility's gap among these three schemes will decrease as the network channel state goes better. These also can be validated by Fig. 5. With respect to rebuffering penalty, as the network state goes better, the QoE gap among these three scheme is increasing. This indicates that the two baseline schemes aggressively request video segment with high bitrates exceeding the network bandwidth. These might lead to more rebuffering events. In other words, our scheme achieve a better balance between bitrate utility and rebuffering penalty than other two baseline schemes.

Finally, we evaluate the performance of the general QoE metric defined as (3). A normalized QoE metric is defined using the min-max normalization method that mapping the original QoE value to the new data between 0 and 1. Figure 8 shows the Cumulative Distribution Function (CDF) of the normalized QoE value across three different channel conditions. There are two key points from these results. First, it can be seen that the percentage of higher normalized QoE values achieved by our scheme is higher than the baseline schemes in all three channel conditions. Second, our scheme outperforms two baseline schemes (MPQUIC and SSIM-CL-w-ERDO) with an improvement in average normalized QoE of 6.5%, 8.3%, 10.2% in the channel condition of average SINR $\bar{\gamma} = 4dB$, $\bar{\gamma} = 9dB$, and $\bar{\gamma} = 14dB$, respectively.

5 Conclusion and limitation

In this paper, a cross-layer DASH-based multipath video streaming scheme is proposed to improve the performance of video streaming. Two wireless access networks, LTE and 802.11ac downlink, are utilized to achieve the bandwidth aggregation. Meanwhile, the cross-layer method is combined with the multipath video streaming by optimizing the MCS modes at the physical layer in each network, the video bitrate, the playback buffering and the bitrate allocation for each segment at the application layer. Experimental results show that our proposed scheme outperformed other state-of-the-art schemes in term of PSNR, playback smoothness and normalized QoE.

In contrast to MPQUIC that runs on top of UDP, the video segment download time of our scheme is a little longer. This mainly is attributed to the TCP protocol used by our scheme or other MPTCP-based approach which will introduce more delay by the acknowledgement mechanism. In the further work, we will focus on the scheduling algorithm for multipath video streaming over MPQUIC in order to further improve the video streaming performance.

Acknowledgements This work was supported in part by National Natural Science Foundation of China under Grant 61771469 and Ningbo Natural Science Foundation under Grant 2019A610109.

References

1. Andrews JG, Buzzi S, Choi W, Hanly SV, Lozano A, Soong AC, Zhang JC (2014) What will 5G be? IEEE JSAC 32(6):1065–1082
2. Argyriou A, Kosmanos D, Tassiulas L (2015) Joint time-domain resource partitioning, rate allocation, and video quality adaptation in heterogeneous cellular networks. IEEE Trans Multimed 17(5):736–745
3. Bruneau QJ, Lacaud M, Negru D, Batalla JM, Borcoci E (2018) Adding a new dimension to HTTP Adaptive Streaming through multiple-source capabilities. IEEE MultiMedia 25(3):65–78
4. Caire G, Muller RR, Knopp R (2007) Hard fairness versus proportional fairness in wireless communications: The single-cell case. IEEE Trans Inform Theory 53(4):1366–1385
5. Campanile L (2020) The network simulator ns-3. http://www.isi.edu/nsnam/ns
6. Chang CY, Yen HC, Lin CC, Deng DJ (2015) Qos/qoe support for H. 264/AVC video stream in IEEE 802.11 ac WLANs. IEEE Syst J 11(4):2546–2555
7. Chen YC, Towsley D, Khalili R (2016) MSPLayer: Multi-source and multi-path video streaming. IEEE JSAC 34(8):2198–2206
8. Cisco Visual Networking Index (2017) Global Mobile Data Traffic Forecast Update, White Paper, 2017-2022, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html
9. Daldoul Y, Meddour DE, Ksentini A (2017) IEEE 802.11 Ac: Effect of channel bonding on spectrum utilization in dense environments. In Proc of IEEE Conf ICC 1–6
10. Deng Z, Liu Y, Liu J, Chen X, Argyriou A, Xu Z, Ci S (2016) Cross-network and cross-layer optimized video streaming over LTE and WCDMA downlink. In Proc of IEEE Conf ISCC 868–873
11. Elgabli A, Aggarwal V (2019) SmartStreamer: Preference-aware multipath video streaming over MPTCP. IEEE Trans Vehicular Technol 68(7):6975–6984
12. Evensen K, Kupka T, Kaspar D, Halvorsen P, Griwodz C (2010) Quality-adaptive scheduling for live streaming over multiple access networks. In Proc of ACM Conf NOSSDAV 21–26
13. Han B, Qian F, Ji L, Gopalakrishnan V (2016) MP-DASH: Adaptive Video streaming over preference-aware multipath. In Proc of ACM Conf CoNEXT 129–143
14. Ho D, Park GS, Song H (2017) Game-theoretic scalable offloading for video streaming services over LTE and WiFi networks. IEEE Trans Mob Comput 17(5):1090–1104
15. Huang TY, Johari R, McKeown N, Trunnell M, Watson M (2014) A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In Proc of ACM Conf SIGCOMM 187–198
16. James C, Halepovic E, Wang M, Jana R, Shankaranarayanan NK (2016) Is multipath TCP (MPTCP) beneficial for video streaming over DASH? in Proc of IEEE Conf MASCOTS, 331–336
17. Jensen TL, Kant S, Wehinger J, Fleury BH (2010) Fast link adaptation for MIMO OFDM. IEEE Trans Vehicul Technol 59(8):3766–3778
18. Jiang J, Sekar V, Zhang H (2014) Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. IEEE/ACM Trans Netw 22(1):326–340
19. Koo J, Yi J, Kim J, Hoque MA, Choi S (2018) Seamless Dynamic Adaptive Streaming in LTE/wi-fi Integrated Network under Smartphone Resource Constraints. IEEE Trans on Mobile Computing 18(7):1647–1660
20. Mao H, Netraval R, Alizadeh M (2017) Neural adaptive video streaming with pensieve. In Proc of ACM Conf SIGCOMM 197–210
21. Ong EH, Kneckt J, Alanen O, Chang Z, Huovinen T, Nihtila T (2011) IEEE 802.11 Ac: Enhancements for very high throughput WLANs. In Proc of IEEE Conf PIMRC 849–853

22. Sesia S, Toufik I, Baker M (2011) LTE-the UMTS long term evolution: From theory to practice. Wiley, New Jersey
23. Spiteri K, Urgaonkar R, Sitaraman RK (2020) BOLA: Near-optimal bitrate adaptation for online videos. IEEE/ACM Trans on Networking. https://doi.org/10.1109/TNET.2020.2996964
24. Stockhammer T (2011) Dynamic adaptive streaming over HTTP– standards and design principles. In Proc of ACM Conf MMSys 133–144
25. Sun Y, Yin X, Jiang J, Sekar V, Lin F, Wang N, Sinopoli B (2016) CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction. In Proc of ACM Conf SIGCOMM 272–285
26. Taranetz M, Blazek T, Kropfreiter T, Muller MK, Schwarz S, Rupp M (2015) Runtime precoding: Enabling multipoint transmission in LTE-advanced system-level simulations. IEEE Access 3:725–736
27. Tourapis AM (2020) H.264/MPEG-4AVC reference software. http://iphome.hhi.de/suehring/tml/download/
28. Viernickel T, Froemmgen A, Rizk A, Koldehofe B, Steinmetz R (2018) Multipath QUIC: A deployable multipath transport protocol. In Proc of IEEE Conf ICC 1–7
29. Xing M, Xiang S, Cai L (2014) A real-time adaptive algorithm for video streaming over multiple wireless access networks. IEEE JSAC 32(4):795–805
30. Yazid M, Ksentini A (2018) Modeling and Performance Analysis of the Main MAC and PHY Features of the 802.11 ac standard: a-MPDU Aggregation vs Spatial Multiplexing. IEEE Trans Vehicul Technol 67(11):10243–10257
31. Yin X, Jindal A, Sekar V, Sinopoli B (2015) A control-theoretic approach for dynamic adaptive video streaming over HTTP. In Proc of ACM Conf SIGCOMM 325–338
32. Yoon D, Cho K, Lee J (2000) Bit error probability of M-ary quadrature amplitude modulation. In: vehicular technology conference fall 2000. in Proc of IEEE Conf VTS Fall, pp. 2422–2427
33. Zhao P, Liu Y, Liu J, Argyriou A, Ci S (2016) SSIM-Based error-resilient cross-layer optimization for wireless video streaming. Signal Process Image Commun 40:36–51