Model Predictive Control Algorithm for Video Coding and Uplink Delivery in Delay-Critical Applications

Mourad Aklouf, Frédéric Dufaux, Michel Kieffer, Marc Lény

IEEE Open Journal of Signal Processing

Received 17 February 2025; revised 26 May 2025; accepted 18 June 2025. Date of publication 30 June 2025; date of current version 29 July 2025. The review of this article was arranged by Associate Editor Bruno Zatt.

Digital Object Identifier 10.1109/OJSP.2025.3584672

Model Predictive Control Algorithm for Video Coding and Uplink Delivery in Delay-Critical Applications

MOURAD AKLOUF [1], FRÉDÉRIC DUFAUX [1] (Fellow, IEEE), MICHEL KIEFFER [1] (Senior Member, IEEE), AND MARC LÉNY [2]

[1]Université Paris-Saclay, CentraleSupélec, CNRS, Laboratoire des Signaux et Systèmes, 91192 Gif-sur-Yvette, France
[2]EKTACOM, F-91940 Les Ulis, France

CORRESPONDING AUTHOR: MICHEL KIEFFER (e-mail: michel.kieffer@l2s.centralesupelec.fr).

This work was supported in part by SMART-V2I and in part by ANR ZL-VLC Project, under Grant ANR-20-CE25-0014.

ABSTRACT Emerging applications such as remote car driving, drone control, or distant mobile robot operation impose a very tight constraint on the delay between the acquisition of a video frame by a camera embedded in the operated device and its display at the remote controller. This paper introduces a new frame-level video encoder rate control technique for ultra-low-latency video coding and delivery. A Model Predictive Control approach, exploiting the buffer level at the transmitter and an estimate of the transmission rate, is used to determine the target encoding rate of each video frame to adapt with minimum delay to sudden variations of the transmission channel characteristics. Then, an $R-(QP, D)$ model of the rate R of the current frame to be encoded as a function of its quantization parameter (QP) and of the distortion D of the reference frame is used to get the QP matching the target rate. This QP is then fed to the video coder. The proposed approach is compared to reference algorithms, namely PANDA, FESTIVE, BBA, and BOLA, some of which have been adapted to the considered server-driven low-latency coding and transmission scenario. Simulation results based on 4G bandwidth traces show that the proposed algorithm outperforms the others at different glass-to-glass delay constraints, considering several video quality metrics.

INDEX TERMS Ultra-low latency, rate adaptation, quality of experience, model predictive control.

I. INTRODUCTION

Emerging applications such as remote car driving [1], remote train driving [2], drone control, or distant mobile robot manipulation [3], require the acquisition, coding, and transmission of video by the controlled device to a remote operator with very strong latency constraints [4], [5], [6]. In such remote control applications, the video frames are acquired by a camera embedded in the controlled device and encoded on-the-fly. The resulting packets are transmitted in uplink direction over a wireless communication channel followed by access and core networks until they reach the remote operator. Packets are then decoded and the resulting video frames are buffered before being displayed to the remote operator. The latter can then take control decisions fed back in downlink direction to the controlled device.

Ultra-Low Latency Video Coding and Delivery (UL-LVCD) algorithms [7], [8], [9] are thus required. Latency constraints differ significantly from classical Downlink Video Streaming (DVS) applications, e.g., to watch a movie on a smartphone. Many DVS applications use HTTP Adaptive Streaming (HAS) [10] to fetch video sequences divided into segments of few seconds, pre-encoded at several bitrates, and stored on servers. The DVS client uses HAS to request video segments with the encoding rate determined by a Streaming Rate Adaptation (SRA) algorithm [11], [12], [13], [14], [15]. They are usually implemented at the client and exploit instantaneous or averaged measurements of the network and channel characteristics and/or of the client buffer level. Their aim is to maximize the Quality of Experience (QoE) of the client (average video quality, frame freezes, number of video quality switches...). In addition to the video coding rate, the playback speed may also be controlled, e.g., using deep Reinforcement Learning (RL) as in [16]. More recently, Chunked transfer encoding (CTE) with MPEG Common Media

TABLE 1. Main Characteristics of DVS and ULLVCD.

	DVS [23]	ULLVCD [2]
Client	Smartphone	Remote operator
Server	Within network	Controlled device
Delay constraint	1-5 s	10-200 ms
Video coding	Offline	Online (at server)
Buffers	Large	Very small
Control type	Streaming rate	Encoding parameters
Controller location	At client	At server
Rate update period	300 ms - 2 s	20 - 40 ms

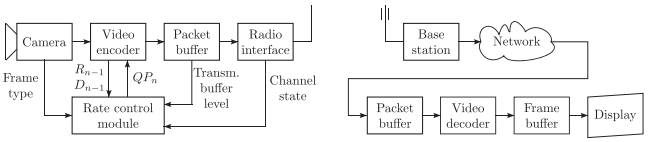

FIGURE 1. Considered server-driven ULLVCD architecture: Transmitter (left) and receiver (right).

Application Format (CMAF) [17] has been proposed as the standard packager for low-latency delivery. CTE allows delivery of a segments in small pieces called chunks, which may be as small as a single video frame. CMAF has been exploited, for example, by Vabis [18], a server-side rate adaptation based on RL for low-latency DVS applications. Real-time Messaging Protocols (RTMP) [19] or Real-time Streaming Protocols (RTSP) [20] associated to Real-time Transport Protocol (RTP) [21] can also be used for low-latency video streaming. With RTMP or RTSP, few milliseconds of video are processed in each packet, making it possible to achieve a transmission latency of about 150 ms [22].

In classical DVS applications, video encoding and segment streaming rate selection are thus performed *separately* and *asynchronously*. In contrast, the delay constraints imposed on ULLVCD algorithms, as well as the high variability of the wireless channel characteristics due to the mobility of the controlled device, impose coding and delivery to be optimized *jointly* within the controlled device. The aim is to minimize the delay between the acquisition of a frame by the controlled device and its display to the remote operator (*glass-to-glass (G2G) delay* [24]). Consequently, large buffers implemented at the DVS client to mitigate variations of the download rate cannot be used by ULLVCD algorithms.

Besides, in classical DVS applications, streaming rate selection is performed with a period equal to the video segment or chunk duration. When the network rate between the server and the client decreases suddenly, the download delay increases and mismatched streaming rate decision lead to a depletion of the client buffer and possibly to a freeze of the displayed frames. In contrast, due to absence of large buffers in ULLVCD applications, video encoding rate has to be decided at the *frame* level and *within* the controlled device to adapt the video encoding rate with minimal delay to the instantaneous rate available on the communication channel. Table 1 compares the main characteristics of *i*) a DVS application used within a smartphone to watch a movie [23] and *ii*) a ULLVCD algorithm used for remote control [2]. It clearly shows that the latter application is much more challenging.

In the context of ULLVCD, this paper proposes a new frame-level encoding rate adaptation algorithm targeting G2G delays between the camera embedded in a controlled device (server) and the display panel of a remote operator (client) between 100 and 200 ms. The proposed algorithm performs a *server-side* control of the *playback margin* of the client, i.e., the difference between the time instant a frame is ready to be displayed and the time instant it is actually displayed. Using measurements of the server buffer level and the channel and network characteristics, we propose a new frame-level Model Predictive Control (MPC) approach [25], [26] to evaluate the target encoding rate of the current frame so as to reach the target playback margin of the client. In the context of our frame-level control, the encoding rate depends on the Quantization Parameter (QP) chosen for the current frame and on the distortion of previously encoded frames, which is not the case in [26]. To address this difficulty, the rate model introduced in [27] is used to predict the size of the current encoded frame as a function of the distortion of the previous frame and of the chosen QP, allowing an efficient selection of the appropriate QP value for each frame to encode. Frame-level MPC of the encoding parameters has been introduced in [25], but without considering strict G2G delay constraints.

The main contributions of this paper consist in

i) the design of a new MPC algorithm for frame-level encoding parameter adaptation using channel and transmission buffer level measurements;
ii) exploiting the R-(QP,D) model in [27] to get frame QP values from encoding rate targets accounting for the distortion of previous frames;
iii) the comparison of the proposed approach to alternative video delivery algorithms [11], [12], [13], [14], some of which have been adapted to meet G2G delay constraints between 100 and 200 ms.

The remainder of this paper is structured as follows. Section II presents an overview of the proposed encoding rate control approach to be embedded in an ULLVCD architecture. The proposed MPC approach for the encoding rate is described in Section III. Section IV recalls the R-(QP, D) model and introduces the way its parameters are iteratively estimated. Simulation results considering real 4G bandwidth traces are reported in Section VI. Finally, some conclusions are drawn in Section VII.

II. OVERVIEW OF THE PROPOSED APPROACH

Fig. 1 shows the components of the proposed encoding rate control algorithm in a server-driven ULLVCD architecture. We consider a single video stream acquired, compressed, and transmitted to a single client. The server consists of a camera, a video encoder, an encoding rate controller, a transmission buffer, and a transmitter. The client includes a packet buffer, a video decoder, a decoded frame buffer, and a display device.

TABLE 2. Main Notations

T_{f}	frame period
$T_{\mathrm{a},n}$, T_{a}	frame acquisition delay, upper bound
$T_{\mathrm{e},n}$, T_{e}	frame encoding delay, upper bound
$T_{\mathrm{d},n}$, T_{d}	frame decoding delay, upper bound
Δ_{p}	acquisition-to-playback delay
τ^*	target client playback margin
$C(t)$	channel rate at time t
$t_n = nT_{\mathrm{f}}$	start of acquisition of frame n
B_n	buffer level in bits at transmitter at t_n
R_n	encoding rate for frame n
$T_{\mathrm{b},n}$	delay to flush B_n bits (frames before frame n)
$T_{\mathrm{r},n}$	delay to flush the encoded bits of frame n
$T_{\mathrm{c},n}$	access and core network delay for frame n
τ_n	client playback margin for frame n

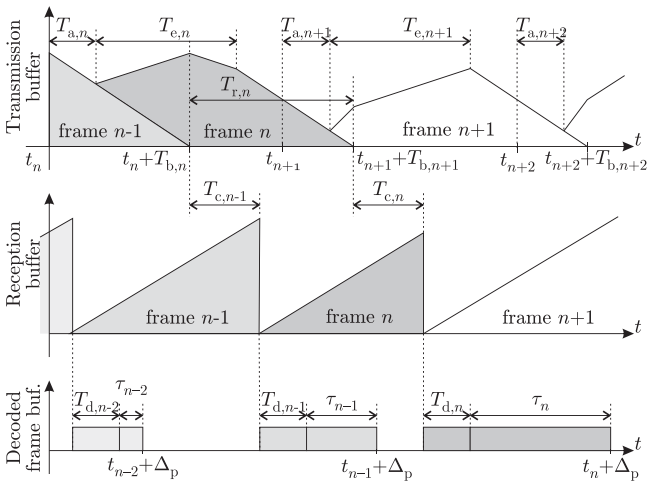

FIGURE 2. Key time instants related to the encoding and transmission of frame *n*: transmission buffer level (top), packet reception buffer level (middle), and decoded frame buffer (bottom).

Table 2 summarizes the main notations used in this paper and Fig. 2 illustrates the different time instants related to the acquisition, processing, transmission, and display of frame n.

We assume that video frames are acquired with a period T_{f}. The acquisition of frame n by the video camera starts at time $t_n = nT_{\mathrm{f}}$. The acquisition delay $T_{\mathrm{a},n}$ depends on the camera shutter type, camera aperture, and scene illumination. The frame acquisition delay can be measured during a camera calibration phase, using techniques presented in [28]. Here, we assume that this delay is upper bounded by T_{a} with $T_{\mathrm{a}} < T_{\mathrm{f}}$, which is consistent with the results presented in [28] for several tested video cameras.

Once frame n is acquired, it is encoded. In this paper, we consider more specifically High-Efficiency Video Coding (HEVC) [29], but other encoders may be used. The trade-off between encoding rate and quality is adjusted by the rate control module via the quantization parameter QP_n for frame n. The encoding delay $T_{\mathrm{e},n}$ is assumed upper bounded by $T_{\mathrm{e}} < T_{\mathrm{f}}$. This constraint can be satisfied considering encoding complexity and delay control mechanisms such as those presented in [30], [31].

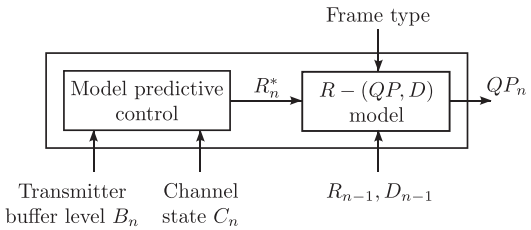

FIGURE 3. Components of the rate control module.

The resulting bitstream is segmented into RTP packets and put into the transmission buffer.

The packets are drained from the buffer and transmitted via 4G or 5G to some base station [32] with a rate $C(t)$ corresponding to the wireless channel capacity. The transmitter has to wait for wireless resources to be granted by the base station to send encoded video packets [33]. For frame n, the buffering delay consists in (*i*) the delay $T_{\mathrm{b},n}$ to flush the bits related to previously encoded frames still present in the buffer and (*ii*) the delay $T_{\mathrm{r},n}$ to drain the bits of frame n. The buffering delay depends thus on the load of the buffer and the amount of uplink resources allocated to the transmitter (which depends on the load of the radio head and its scheduling policy and impacts directly $C(t)$).

Packets are then carried out through the access and core networks to the receiver. This introduces a delay $T_{\mathrm{c},n}$ which depends on the congestion of routers along the path between the base station and the receiver, as well as on the length of this path. The channel capacity $C(t)$ is impacted by fading, interference, and mobility and can therefore vary within few milliseconds [34] in applications such as remote drone, car, or train control. In contrast, the propagation delay $T_{\mathrm{c},n}$ evolves slower with time. Consequently, in what follows, $T_{\mathrm{c},n}$ is assumed efficiently predicted, e.g., using techniques such as [35], [36].

At the client side, we assume that decoding only starts once all packets related to frame n have been received. This introduces a decoding delay $T_{\mathrm{d},n}$ assumed upper bounded by $T_{\mathrm{d}} < T_{\mathrm{f}}$. The resulting frames are then temporarily buffered before being displayed at time $t_n + \Delta_{\mathrm{p}}$, where Δ_{p} is the acquisition-to-playback delay or G2G delay.

The control is performed with the goal to ensure that frames are always displayed on time. If a frame is not decodable, e.g., due to packet loss or corruption, a frame concealment process is applied [37]. Outdated packets in the transmission buffer, i.e., packets which have no chance to reach the receiver on time are purged, as suggested in [13].

To determine the value of the quantization parameter QP_n for frame n, the rate control module in Fig. 3 takes as input the amount of bits B_n stored in the transmission buffer as well as an estimate $\widehat{C}(t)$ of the channel rate $C(t)$, e.g., obtained using the approach proposed in [38], [39], see also [40]. This

estimate is used by the MPC algorithm to determine the rate at which the packets will be transmitted over the wireless channel during the time interval $[t_n, t_n + \Delta_p]$ and to evaluate the delivery delay for each frame n. The MPC determines then the target encoding rate R_n^* of the frame n to reach a playback margin τ^* at the receiver.

In this paper, a low-delay P-frame configuration is considered, with a periodic succession of one I-frame and several predicted P-frames. The encoding rate of I-frames is adapted to that of P-frames to avoid a large rate jitter leading to a delay jitter due to the increased transmission latency for I-frames and the following P-frames. Alternatively, a gradual decoding refresh (GDR) configuration can be considered [41], where each frame contains a part encoded in Intra mode, the remainder of the frame being encoded in Inter mode. In both configurations (low-delay P or GDR), since each P-frame is encoded considering the previous frame as reference, the relation between R_n and QP_n largely depends on the distortion D_{n-1} of the reference frame $n-1$ used to encode frame n. The R-(QP, D) model proposed in [27] provides the rate of frame n as a function of the distortion D_{n-1} of the frame $n-1$ and of QP_n. This model is used to select the value of QP_n to encode frame n so that R_n is as close as possible to R_n^*. An adjustment at the CTU level of the encoding parameters may be performed [42], [43] to reach more accurately the rate target R_n^*. This requires, however, an additional rate control embedded inside the video coder, while in our approach, only the QP of each frame is provided.

Since the temporal and spatial characteristics of the frames evolve with time, an update of the parameters of the R-(QP, D) model has to be performed online. Section IV recalls the structure of the R-(QP, D) model and details the way its parameters may be estimated iteratively.

III. FRAME-LEVEL MPC OF THE ENCODING RATE

Our aim with the considered MPC approach is to determine the target encoding rates $R_{n+1}^*, \ldots, R_{n+h}^*$ of frames $n+1, \ldots, n+h$ before the encoding start time $t_{n+1} + T_{a,n+1}$ of frame $n+1$, where h is the prediction horizon. For this purpose, a model of the channel capacity is used to predict the impact of the chosen encoding rates on the playback margins $\tau_{n+1}, \ldots, \tau_{n+h}$. The target encoding rates are chosen so that the playback margins meet some target τ^*

$$\left(R_{n+1}^*, \ldots, R_{n+h}^*\right) = \underset{(R_{n+1},\ldots,R_{n+h})}{\arg\min} \sum_{i=1}^{h} \left(\tau_{n+i} - \tau^*\right)^2. \quad (1)$$

Then R_{n+1}^* is selected and the next target encoding rates $R_{n+2}^*, \ldots, R_{n+1+h}^*$ are evaluated during the time interval $[t_{n+1} + T_{a,n+1}, t_{n+2} + T_{a,n+2}]$. Such farsighted evaluation of the encoding rates over an horizon $h > 1$ allows a smoother control of the variations of the encoding rates. For example, if the future channel rate is expected to decrease after time t_{n+2}, the encoding rate for frame $n+1$ may be decreased by anticipation to avoid an abrupt decrease of the encoding rate of frames $n+2$, $n+3$...

In what follows, first, the playback margin of frame n is evaluated considering the buffer level B_n at time t_n and the chosen encoding rate R_n for frame n, see Section III-A. Then, in Section III-B, to determine the encoding rates $R_{n+1}^*, \ldots, R_{n+h}^*$ using (1), the playback margins $\tau_{n+1}, \ldots, \tau_{n+h}$ are evaluated iteratively as a function of $R_{n+1}, \ldots, R_{n+h}$ using the information available at time t_n, i.e., B_n and R_n, as well as estimates of the future channel capacities. This allows resorting to a numerical optimization to get $R_{n+1}^*, \ldots, R_{n+h}^*$. Finally, when considering $h = 1$ and some additional simplifying assumptions, an approximate explicit expression of R_{n+1}^* is obtained in Section III-C.

In all derivations, the upper bounds for the acquisition, encoding, and decoding delays have been considered. These upper bounds may be replaced in the derivations by online estimates of the delays, obtained using techniques such as those presented in [44].

A. PLAYBACK MARGIN OF FRAME n

At time t_n, we assume that the transmission buffer contains B_n bits from packets related to previously encoded frames. Considering the channel rate $C(t)$ at time t, the time $T_{b,n}$ required to flush these B_n bits from the transmission buffer satisfies (see Fig. 2 (top))

$$B_n = \int_{t_n}^{t_n + T_{b,n}} C(t)dt. \quad (2)$$

With an encoding rate R_n for frame n, $R_n T_f$ bits are generated during the encoding process. We consider that the transmission buffer is fed at a constant rate larger than the channel capacity over the encoding time interval, i.e., that $R_n T_f / T_{e,n} > C(t), \forall t \in [t_n + T_{a,n},\ t_n + T_{a,n} + T_{e,n}]$. Consequently, the transmission buffer does not get empty over the encoding time interval. Hence, the delay $T_{r,n}$ required to drain the $R_n T_f$ bits of the encoded frame n from the transmission buffer satisfies

$$R_n T_f = \begin{cases} \int_{t_n + T_{b,n}}^{t_n + T_{b,n} + T_{r,n}} C(t)dt & \text{if } T_{b,n} \geqslant T_{a,n}, \\ \int_{t_n + T_{a,n}}^{t_n + T_{a,n} + T_{r,n}} C(t)dt & \text{if } T_{b,n} < T_{a,n}. \end{cases} \quad (3)$$

The first case in (3) corresponds to a transmission buffer still containing bits when the video encoder starts to feed bits from frame n, as illustrated in Fig. 2 (top). In the second case, all bits from previous frames have been drained before time $t_n + T_{a,n}$.

Then, the last bit from frame n reaches the packet buffer at the receiver side with a delay $T_{c,n}$ after its transmission. Accordingly, it is received at time $t_n + \max\{T_{a,n}, T_{b,n}\} + T_{r,n} + T_{c,n}$, see Fig. 2 (middle). The packets are then decoded, with a delay $T_{d,n}$, to get a frame n ready to be displayed by the receiver at time $t_n + \max\{T_{a,n}, T_{b,n}\} + T_{r,n} + T_{c,n} + T_{d,n}$, see Fig. 2 (bottom). Since frame n is actually displayed at time $t_n + \Delta_p$, the playback margin for frame n is

$$\begin{aligned} \tau_n &= t_n + \Delta_p - \left(t_n + \max\left\{T_{a,n}, T_{b,n}\right\} + T_{r,n} + T_{c,n} + T_{d,n}\right) \\ &= \Delta_p - \left(\max\left\{T_{a,n}, T_{b,n}\right\} + T_{r,n} + T_{c,n} + T_{d,n}\right). \end{aligned} \quad (4)$$

Considering the upper bounds for $T_{a,n}$ and $T_{d,n}$, a lower bound $\underline{\tau}_n$ for the playback margin of frame n is obtained from (4) as

$$\underline{\tau}_n = \Delta_p - \left(\max\left\{T_a, T_{b,n}\right\} + T_{r,n} + T_{c,n} + T_d\right). \quad (5)$$

B. PLAYBACK MARGINS OF FRAMES $n+1, \ldots, n+h$

In this section, the playback margins $\tau_{n+1}, \ldots, \tau_{n+h}$ are evaluated by induction as functions of $R_{n+1}, \ldots, R_{n+h}$ for a given value of R_n and B_n. For that purpose, we assume that $C(t)$ is piecewise constant and equal to C_n over time intervals of the form $[t_n, t_n + T_f]$ for all $n > 0$.

We start with an evaluation of τ_{n+1} as a function of R_{n+1}. This requires an evaluation of $T_{b,n}$ and of $T_{r,n}$, detailed in what follows.

If all B_n bits are drained from the transmission buffer over the time interval $[t_n, t_n + T_f]$, as is the case in Fig. 2 (top), using (2) one gets

$$T_{b,n} = B_n / C_n, \quad (6)$$

as the channel capacity over $[t_n, t_n + T_f]$ is C_n. This should be the normal situation in an ULLVCD context where the main target is low latency. If more time is required, the variations of the channel capacity have to be taken into account in (2). Consider $\ell_{b,n}$, the number of time intervals of duration T_f over which the bits present in the transmission buffer at time t_n are drained. Then $\ell_{b,n}$ is the largest integer such that

$$B_n - \sum_{\ell=0}^{\ell_{b,n}-1} C_{n+\ell} T_f > 0. \quad (7)$$

The last bits from B_n are flushed during the time interval $[t_n + \ell_{b,n} T_f, t_n + (\ell_{b,n}+1)T_f]$. One has $\ell_{b,n} = \lfloor T_{b,n}/T_f \rfloor$, where $\lfloor \cdot \rfloor$ denotes downwards rounding. Now, decomposing the integral in (2) in intervals over which the capacity $C(t)$ is constant, one gets

$$B_n = (T_{b,n} - \ell_{b,n} T_f) C_{n+\ell_{b,n}} + \sum_{\ell=0}^{\ell_{b,n}-1} C_{n+\ell} T_f \quad (8)$$

from which $T_{b,n}$ is obtained as

$$T_{b,n} = \ell_{b,n} T_f + \left(B_n - \sum_{\ell=0}^{\ell_{b,n}-1} C_{n+\ell} T_f\right) / C_{n+\ell_{b,n}} \quad (9)$$

To evaluate $T_{r,n}$, a similar approach is considered. If the $R_n T_f$ bits from frame n are drained during the interval $[t_n + \max\{T_{a,n}, T_{b,n}\}, t_n + (\ell_{b,n}+1)T_f]$, (3) leads to

$$T_{r,n} = R_n T_f / C_{n+\ell_{b,n}}. \quad (10)$$

When more time is required, let $\ell_{r,n}$ be the number of time intervals of duration T_f during which the $R_n T_f$ bits of frame n are drained from the transmission buffer. The last of these bits are drained during the time interval $[t_n + (\ell_{b,n} + \ell_{r,n})T_f,\ t_n + (\ell_{b,n} + \ell_{r,n} + 1)T_f]$. As in (7), $\ell_{r,n}$ is the largest integer such that $R_n T_f - C_{n+\ell_{b,n}}((\ell_{b,n}+1)T_f - T_{b,n}) - \sum_{\ell=\ell_{b,n}+1}^{\ell_{b,n}+\ell_{r,n}-1} C_{n+\ell} T_f > 0$. To determine $T_{r,n}$, the integral in (3) is decomposed over time intervals for which $C(t)$ is constant to get

$$\begin{aligned} R_n T_f = {} & C_{n+\ell_{b,n}} \left((\ell_{b,n}+1) T_f - \max\left\{T_{a,n}, T_{b,n}\right\}\right) \\ & + \textstyle\sum_{\ell=\ell_{b,n}+1}^{\ell_{b,n}+\ell_{r,n}-1} C_{n+\ell} T_f \\ & + \left(T_{r,n} - \left(\ell_{b,n} + \ell_{r,n}\right) T_f\right) C_{n+\ell_{b,n}+\ell_{r,n}}, \end{aligned} \quad (11)$$

The first term of (11) is the amount of bits drained during the time interval $[t_n + \max\{T_{a,n}, T_{b,n}\}, t_n + (\ell_{b,n}+1)T_f]$. The second term represents the amount of bits drained during the $\ell_{r,n} - 1$ following time intervals. The third term is the amount of bits drained during the last time interval $[t_n + (\ell_{b,n} + \ell_{r,n})T_f, t_n + (\ell_{b,n} + \ell_{r,n} + 1)T_f]$. The expression of $T_{r,n}$ is then deduced from (11) as

$$\begin{aligned} T_{r,n} = {} & \left(\ell_{b,n} + \ell_{r,n}\right) T_f \\ & + \Big(R_n T_f - C_{n+\ell_{b,n}} \left((\ell_{b,n}+1) T_f - \max\left\{T_{a,n}, T_{b,n}\right\}\right) \\ & + \textstyle\sum_{\ell=\ell_{b,n}+1}^{\ell_{b,n}+\ell_{r,n}-1} C_{n+\ell} T_f \Big) / C_{n+\ell_{b,n}+\ell_{r,n}}. \end{aligned} \quad (12)$$

The time instant $t_n + T_{r,n}$ is also equal to the time instant $t_{n+1} + T_{b,n+1}$ at which the bits related to frames with an index strictly less than $n+1$ are drained from the transmission buffer. As $t_{n+1} = t_n + T_f$, one has

$$T_{b,n+1} = T_{r,n} - T_f. \quad (13)$$

Note that $T_{b,n+1}$ may be negative when all bits related to frame n are flushed before t_{n+1}. Moreover

$$\begin{aligned} \ell_{b,n+1} &= \lfloor \left(T_{r,n} - T_f\right) / T_f \rfloor \\ &= \lfloor T_{r,n}/T_f \rfloor - 1. \end{aligned}$$

The transmission of the bits from frame $n+1$, encoded at a rate R_{n+1}, starts at $t_{n+1} + \max\{T_{a,n+1}, T_{b,n+1}\}$ and ends at time $t_{n+1} + T_{r,n+1}$, where $T_{r,n+1}$ may be written as (10) or as (12), depending on the amount of bits to transmit, by replacing n by $n+1$. Then, the playback margin for frame $n+1$ is deduced from (4) as

$$\begin{aligned} \tau_{n+1} &= \Delta_p - \big(\max\left\{T_{a,n+1}, T_{b,n+1}\right\} \\ &\qquad + T_{r,n+1} + T_{c,n+1} + T_{d,n+1}\big) \\ &= \Delta_p - \big(\max\left\{T_{a,n+1}, T_{r,n} - T_f\right\} \\ &\qquad + T_{r,n+1} + T_{c,n+1} + T_{d,n+1}\big), \end{aligned}$$

where $T_{r,n+1}$ depends on R_{n+1}. A lower bound $\underline{\tau}_{n+1}$ of τ_{n+1} is

$$\begin{aligned} \underline{\tau}_{n+1} = \Delta_p - {} & \big(\max\left\{T_a, T_{r,n} - T_f\right\} \\ & + T_{r,n+1} + T_{c,n+1} + T_d\big). \end{aligned} \quad (14)$$

The playback margins $\tau_{n+2}, \ldots, \tau_{n+h}$ are evaluated similarly by induction. For example $t_{n+1} + T_{r,n+1}$ is equal to $t_{n+2} + T_{b,n+2}$ and $T_{r,n+2}$ can be estimated for a given value of R_{n+2} provided that estimates of the future channel capacities are available. The playback margin τ_{n+2} can then be evaluated as a function of R_{n+2}, R_{n+1}, R_n, B_n, and the (estimated) channel capacities.

Consequently, for a given value of B_n, R_n, and assuming that the channel capacities $C_n, \ldots, C_{n+1+\ell_{b,n+1}+\ell_{r,n+1}}$ are either known or estimated, $\underline{\tau}_{n+1}$ can be evaluated as a function of R_{n+1} using (14). Then, when $h = 1$, R^*_{n+1} may be chosen such that $\underline{\tau}_{n+1} = \tau^*$ using, e.g., a search by dichotomy. When $h > 1$, an iterative numerical optimization of (1) has to be performed to obtain $(R^*_{n+1}, \ldots, R^*_{n+h})$.

C. APPROXIMATE EXPRESSION OF THE RATE R_{n+1}

In this section, we focus on the case $h = 1$ to get an explicit expression of R_{n+1}.

When all B_n bits are drained from the transmission buffer over the time interval $[t_n, t_n + T_f]$, $T_{b,n}$ is given by (6). Similarly, if all B_{n+1} bits remaining at t_{n+1} in the transmission buffer are drained over the time interval $[t_{n+1}, t_{n+1} + T_f]$, one has

$$T_{b,n+1} = B_{n+1}/C_{n+1}. \tag{15}$$

Assuming that the channel capacity is fully exploited by the encoding rate control algorithm, and the transmission buffer never gets empty, the transmission buffer level B_{n+1} at time t_{n+1} satisfies

$$B_{n+1} = B_n + (R_n - C_n)\, T_f. \tag{16}$$

The bits related to frame n may be drained over several time intervals, as illustrated in (12) and in Fig. 2 (top). Assuming that the channel capacity does not vary too quickly between frames, (3) may be approximated by

$$T_{r,n} = R_n T_f / C_n \tag{17}$$

for frame n and

$$T_{r,n+1} = R_{n+1} T_f / C_{n+1} \tag{18}$$

for frame $n + 1$.

The evaluation of R_{n+1} starts at $t_n + T_{a,n}$. Using (4) and assuming that $T_{c,n+1} = T_{c,n}$ and $T_{d,n+1} = T_{d,n}$, one gets

$$\begin{aligned}\tau_{n+1} = \tau_n &+ \max\left\{T_{a,n}, T_{b,n}\right\} - \max\left\{T_{a,n+1}, T_{b,n+1}\right\} \\ &+ \left(T_{r,n} - T_{r,n+1}\right).\end{aligned} \tag{19}$$

Introducing (6) and (15)–(18) in (19), one obtains

$$\begin{aligned}\tau_{n+1} = \tau_n &+ B_n/C_n - \left(B_n + (R_n - C_n)\, T_f\right)/C_{n+1} \\ &+ \left(R_n/C_n - R_{n+1}/C_{n+1}\right) T_f.\end{aligned} \tag{20}$$

A minimization of (1) accounting for (20) leads to

$$R^*_{n+1} = \frac{\tau_n - \tau^*}{T_f} C_{n+1} + \left(\frac{C_{n+1}}{C_n} - 1\right)\left(\frac{B_n}{T_f} + R_n\right) + C_n. \tag{21}$$

In practice, R^*_{n+1} given by (21) may not necessarily be positive. In the latter case, $R^*_{n+1} = R_{\min}$, a minimum frame encoding rate is selected to ensure a minimum quality of the received frame.

The evaluation of R^*_{n+1} using (21) is performed at the transmitter side. The playback margin τ_n for frame n is observed at client side at time $t_n + \Delta_p - \tau_n$. Even if R^*_n has been chosen to get a playback margin τ^* for frame n, due to the discrepancies between R^*_n and the actual encoding rate R_n and between the channel capacity estimate $\widehat{C}_n$ used at time $t_{n-1} + T_{a,n-1}$ to evaluate R^*_n and the channel capacity experienced over the time interval $[t_n, t_n + T_f]$, the actual playback margin τ_n is likely to differ from τ^*. Consequently, it is useful to update the estimate $\widehat{\tau}_n$ of τ_n when evaluating R^*_{n+1}.

The proposed approach also uses estimates $\widehat{C}_n$ and $\widehat{C}_{n+1}$ of the channel rates C_n and C_{n+1}. Designing an effective channel rate estimator is essential, but outside the scope of this paper focusing on the encoding rate adaptation. Estimators such as [38], [39] may be exploited. Then, (21) becomes

$$R^*_{n+1} = \frac{\widehat{\tau}_n - \tau^*}{T_f} \widehat{C}_{n+1} + \left(\frac{\widehat{C}_{n+1}}{\widehat{C}_n} - 1\right)\left(\frac{B_n}{T_f} + R_n\right) + \widehat{C}_n \tag{22}$$

with $\widehat{\tau}_n$ obtained introducing (6) and (17) in (4) to get

$$\widehat{\tau}_n = \Delta_p - \left(\left(B_n + R_n T_f\right)/\widehat{C}_n + T_{c,n} + T_d\right). \tag{23}$$

Some insights on (22) may be obtained considering the target number of bits allocated to frame $n + 1$

$$\begin{aligned}R^*_{n+1} T_f = &\left(\widehat{\tau}_n - \tau^*\right)\widehat{C}_{n+1} + \left(\widehat{C}_n T_f - R_n T_f\right) \\ &+ \left(\widehat{C}_{n+1}/\widehat{C}_n - 1\right) B_n + \widehat{C}_{n+1} R_n T_f / \widehat{C}_n\end{aligned} \tag{24}$$

to reach a playback margin equal to τ^*. If $\widehat{\tau}_n > \tau^*$, the estimated playback margin for frame n is larger than the target, and the first term in (24) shows that more bits may be used to represent frame $n + 1$. The second term of (24) indicates that more bits may be used to represent frame $n + 1$ when $\widehat{C}_n T_f > R_n T_f$, i.e. when more bits are drained from the transmission buffer than those fed by the encoding of frame n. The third term of (24) takes into account the increase or decrease of the allowed rate due to a more or less efficient drain of the bits present in the transmission buffer at time t_n. The last term of (24) corresponds to the number of bits to be transmitted in steady-state. If $\widehat{C}_{n+1} > \widehat{C}_n$, the target rate can increase, otherwise it has to decrease.

IV. R-(QP, D) MODEL

In the considered approach, each frame n has to be encoded on-the-fly with encoding parameters adapted to reach the encoding rate target R^*_n provided by (24). This can be achieved by adjusting the coding parameters, e.g., at the frame [25], [45] or at the CTU level [42], [46], [47].

The aim of the R-(D,QP) model considered in this section is to provide QP_n for frame n such that the rate R_n at the output of the coder is as close as possible to the encoding rate target R^*_n, by accounting for the type of frame, and for the distortion D_{n-1} of the frame $n - 1$ serving as reference to encode frame n, see also Fig. 3. This approach does not require any rate adjustment at the CTU level. Nevertheless, it can be complemented by rate control approaches at CTU level, where the provided QP_n may serve as initialization for the coding parameters of CTUs. It may also supplement the previously-mentioned encoding rate control approaches.

In what follows, we consider a low-delay coding configuration (I-frame followed by P-frames, with potentially Gradual Decoder Refresh (GDR) mode activated), where each P-frame uses the last coded frame as reference. Section IV-A recalls the model structure. Section IV-B describes the way its parameters are iteratively estimated. Section IV-C details the estimation of QP_n for a given encoding rate target R_n^*.

A. MODEL STRUCTURE

The dependency of the size $R_n T_f$ for the P-frame n when encoded with QP_n, considering the distortion D_{n-1} of its reference frame, here frame $n-1$, has been modeled in [27] as

$$\begin{aligned} R\left(QP_n, D_{n-1}, \mathbf{p}_n\right) = p_1 \exp\left(-p_2 QP_n\right) \\ + p_3\left(1 - p_4 \log\left(QP_n\right)\right) \\ \times \left(1 + \tanh\left(p_5 QP_n \log\left(D_{n-1}\right)\right.\right. \\ \left.\left. - \left(p_6 QP_n - p_7\right)^2\right)\right). \end{aligned} \tag{25}$$

where $\mathbf{p}_n = (p_{n,1}, \ldots, p_{n,7})$ is the vector of model parameters, which has to be estimated.

The first term of (25) does not depend on D_{n-1} and is a classical rate-distortion model for a whole group of pictures (GOP) or for an I-frame, see [48]. The second term of (25) translates the impact of D_{n-1} on R_n. When D_{n-1} is very small, the hyperbolic tangent is close to -1 and the second term vanishes. R_n only depends on QP_n as the reference frame is of high quality. Large values of D_{n-1} lead to a rate penalty to compensate for the low quality of the reference frame.

B. ITERATIVE PARAMETER ESTIMATION

The parameter $\mathbf{p}_n$ has to be estimated for each frame n. To get enough training data, in [27], each frame is encoded with many different QPs, which is not practical in the ULLVCD context. A reduced-complexity recursive least-square estimation [49] of $\mathbf{p}_n$ from $\mathbf{p}_{n-1}$ is used here and detailed in what follows.

Introducing the difference $\boldsymbol{\delta}_n$ between two successive values of the vector of parameters, $\mathbf{p}_n$ can be expressed as

$$\mathbf{p}_n = \mathbf{p}_{n-1} + \boldsymbol{\delta}_n. \tag{26}$$

Assuming that the rate-distortion characteristics of consecutive encoded video frames of the same GOP vary slowly and smoothly, $\boldsymbol{\delta}_n$ remains small. The first-order Taylor approximation of (25) around the estimate $\widehat{\mathbf{p}}_{n-1}$ of $\mathbf{p}_{n-1}$ is

$$\begin{aligned} \widetilde{R}\left(QP_n, D_{n-1}, \widehat{\mathbf{p}}_{n-1} + \boldsymbol{\delta}_n\right) &= R\left(QP_n, D_{n-1}, \widehat{\mathbf{p}}_{n-1}\right) \\ &+ \frac{\partial R\left(QP_n, D_{n-1}, \widehat{\mathbf{p}}_{n-1}\right)}{\partial \widehat{\mathbf{p}}_{n-1}^T} \boldsymbol{\delta}_n. \end{aligned} \tag{27}$$

Let us consider M video encoders running in parallel. The $M-1$ first ones encode frame n with different $QP_{n,m}$, $m = 1, \ldots, M-1$. The M-th encoder uses the value QP_n^* for frame n deduced from (33) and the output R_n^* of the encoding rate controller. Let $D_{n-1,m}$, $m = 1, \ldots, M$, be the distortion obtained for frame $n-1$ when encoded by the m-th encoder and $R_{n,m}$ be the rate of frame n at the output of the m-th encoder. Using $D_{n-1,m}$, $QP_{n,m}$, and $R_{n,m}$, $m = 1, \ldots, M$, one evaluate the estimate $\widehat{\boldsymbol{\delta}}_n$ of $\boldsymbol{\delta}_n$ that minimize the regularized weighted least-squares cost function

$$\begin{aligned} \widehat{\boldsymbol{\delta}}_n = \arg\min_{\boldsymbol{\delta}} \sum_{m=1}^{M} w_{n,m} \left(R_{n,m}\right. \\ \left. - R\left(QP_{n,m}, D_{n-1,m}, \widehat{\mathbf{p}}_{n-1} + \boldsymbol{\delta}\right)\right)^2 + \alpha \boldsymbol{\delta}^T \boldsymbol{\delta}, \end{aligned} \tag{28}$$

where $w_{n,m} \geqslant 0$, $m = 1, \ldots, M$ are some weights and $\alpha \geqslant 0$ is a regularizing coefficient to favor small values of $\boldsymbol{\delta}$.

Introducing

$$\begin{aligned} \mathbf{y}_n = \left(R_{n,1} - R\left(QP_{n,1}, D_{n-1,1}, \widehat{\boldsymbol{p}}_{n-1}\right), \ldots\right. \\ \left. R_{n,M} - R\left(QP_{n,M}, D_{n-1,M}, \widehat{\boldsymbol{p}}_{n-1}\right)\right)^T, \end{aligned} \tag{29}$$

$$\mathbf{W}_n = \operatorname{diag}\left(w_{n,1}, \ldots, w_{n,M}\right), \tag{30}$$

$$\mathbf{X}_n = \begin{pmatrix} \frac{\partial R\left(Q_{n,1}, D_{n-1,1}, \widehat{\mathbf{p}}_{n-1}\right)}{\partial \widehat{\mathbf{p}}_{n-1}^T} \\ \vdots \\ \frac{\partial R\left(Q_{n,M}, D_{n-1,M}, \widehat{\mathbf{p}}_{n-1}\right)}{\partial \widehat{\mathbf{p}}_{n-1}^T} \end{pmatrix}, \tag{31}$$

and using (27), one may rewrite (28) as

$$\widehat{\boldsymbol{\delta}}_n = \arg\min_{\boldsymbol{\delta}} \left(\mathbf{y}_n - \mathbf{X}_n \boldsymbol{\delta}\right)^T \mathbf{W}_n \left(\mathbf{y}_n - \mathbf{X}_n \boldsymbol{\delta}\right) + \alpha \boldsymbol{\delta}^T \boldsymbol{\delta},$$

from which one obtains

$$\widehat{\boldsymbol{\delta}}_n = \left(\mathbf{X}_n^T \mathbf{W}_n \mathbf{X}_n + \alpha \mathbf{I}\right)^{-1} \mathbf{W}_n \mathbf{X}_n^T \mathbf{y}_n. \tag{32}$$

Then, $\widehat{\mathbf{p}}_n$ is determined from $\widehat{\mathbf{p}}_{n-1}$ and $\widehat{\boldsymbol{\delta}}_n$ as

$$\widehat{\mathbf{p}}_n = \widehat{\mathbf{p}}_{n-1} + \widehat{\boldsymbol{\delta}}_n.$$

The choice of $\mathbf{W}_n$, α and of the different $QP_{n,m}, m = 1, \ldots, M-1$ is discussed in Section VI-A.

The QP of the first frame in the GOP coded as I-frame, can be determined using the first part of (25) or models such as those presented in [48].

Considering $M-1$ extra encoders introduces additional costs in terms of material. Nevertheless, this does not introduce any additional encoding delay, as these encoders can run in parallel. To reduce their memory footprint, the $M-1$ additional coders may share the same decoded frame buffer, but in this case, the $D_{n-1,m}, m = 1, \ldots, M$ would all be equal. This reduces the diversity of input data to train the model (25).

C. EVALUATION OF QP_n

Provided that $\mathbf{p}_n$ and D_{n-1} are known, (25) is used to select the value QP_n^* leading to $R(QP_n^*, D_{n-1}, \mathbf{p}_n)$ closest to R_n^*, i.e.,

$$QP_n^* = \arg\min_{QP \in \mathcal{Q}_n} \left(R\left(QP, D_{n-1}, \mathbf{p}_n\right) - R_n^*\right)^2, \tag{33}$$

where $\mathcal{Q}_n$ is the set of admissible values of QP for frame n. As $\mathcal{Q}_n$ contains a very limited number of elements, QP_n^* may be obtained by exhaustive search with a negligible complexity compared to that of encoding a video frame, as an explicit expression of $R(QP, D_{n-1}, \mathbf{p}_n)$ is available.

Algorithm 1: Rate Control Algorithm.

Initialization: $D_0 = 0$, $R_0^* = R_0$
for $n = 1 : N$ **do**
  % At time t_n
  Acquire frame n
  F =Select_frame_type(n)
  QP_n^* =Select_QP(R_n^*, D_{n-1}, F)
  $(R_{n+1}^*, \ldots, R_{n+h}^*)$ =Optim_encod_rate(B_n, R_n^*, C_n)
  % At time $t_n + T_{\mathrm{a},n}$
  (R_n, D_n) =Encode_frame(F, QP_n^*, R_n^*)

Algorithm 2: Optim_encod_rate.

Input: B_n, R_n^*, C_n
Output: $R_{n+1}^*, \ldots, R_{n+h}^*$
$(\widehat{C}_{n+1}, \ldots, \widehat{C}_{n+L})$ =Predict_Channel_Rate $((C_n, C_{n-1}, \ldots), L)$
$T_{\mathrm{b},n}$ =Eval_flush_delay$(B_n, C_n, (\widehat{C}_{n+1}, \ldots, \widehat{C}_{n+L}))$
$T_{\mathrm{r},n}$ =Eval_drain_delay $(R_n^* T_{\mathrm{F}}, \max(T_{\mathrm{b},n} T_{\mathrm{a}}), C_n, (\widehat{C}_{n+1}, \ldots, \widehat{C}_{n+L}))$
$\widehat{\tau}_n = \Delta_{\mathrm{p}} - (\max(T_{\mathrm{b},n} T_{\mathrm{a}}) + T_{\mathrm{r},n} + T_{\mathrm{c},n} + T_{\mathrm{d}})$
$(R_{n+1}^*, \ldots, R_{n+h}^*)$ =Optimize_playback_margin $(T_{\mathrm{r},n}, (\widehat{C}_{n+1}, \ldots, \widehat{C}_{n+L}))$

Algorithm 3: Eval_playback_margins.

Input: $T_{\mathrm{r},n}, (R_{n+1}, \ldots, R_{n+h}), (\widehat{C}_{n+1}, \ldots, \widehat{C}_{n+L})$
Output: $\tau_{n+1}, \ldots, \tau_{n+h}$
for $\ell = 1 : h$ **do**
  $T_{\mathrm{b},n+i} = T_{\mathrm{r},n+i-1} - T_{\mathrm{f}}$
  $\ell_{\mathrm{b},n+i} = \max\{0, \lfloor T_{\mathrm{r},n+i-1}/T_{\mathrm{f}} \rfloor - 1\}$
  Evaluate $\ell_{\mathrm{r},n+i}$ using (11)
  Evaluate $T_{\mathrm{r},n+i+1}$ using (10) or (12)
  Evaluate $\underline{\tau}_{n+i}$ using (14)

V. RATE CONTROL ALGORITHM

The encoding rate control algorithm is provided in Algorithm 1. At time t_n, several actions are done in parallel. Frame n is acquired, QP_n^* is determined, and the encoding rates $R_{n+1}^*, \ldots, R_{n+h}^*$ are evaluated for a prediction horizon of h steps. At time $t_n + T_{\mathrm{a},n}$, frame n is encoded. In parallel, bits are continuously drained from the transmission buffer.

Algorithm 2 evaluates the encoding rates $R_{n+1}^*, \ldots, R_{n+h}^*$ over a prediction horizon of h steps. The L future channel rates are first evaluated using previous channel rate measurements. The transmitter buffer flush delay $T_{\mathrm{b},n}$ is then determined using (6) or (9). The drain delay $T_{\mathrm{r},n}$ for the encoded bits of frame n is evaluated using (10) or (12). The cost function (1) is then optimized using an iterative search algorithm exploiting the function described in Algorithm 3 which evaluates the playback margins for a given value of $(R_{n+1}, \ldots, R_{n+h})$.

VI. PERFORMANCE EVALUATION

This section compares the proposed MPC approach with reference rate-based and buffer-based schemes.

A. SIMULATION SETUP AND EVALUATION METRICS

The simulation setup consists of a server and a client as described in Section II. The server acquires the video frames, runs the HEVC encoder, and feeds the transmission buffer with encoded packets. It also manages the rate control algorithm, described in Sections III and V and the parameter estimation of the R-(QP, D) model described in Section IV. The client contains a reception buffer, an HEVC decoder, and a decoded frame buffer, see Fig. 2. The access and core networks are simulated using 4G bandwidth traces from [50].

Five video sequences belonging to the JVET test sequences are used: *CrowdRun*, *ParkJoy*, *TouchDownPass*, *DaylightRoad2*, and *KristenandSara* [51]. These video sequence span a wide range of spatial and temporal information. The video sequences are sub-sampled using FFmpeg [52] to two spatial resolutions 640 × 360 and 1280 × 720, and a temporal resolution of 25 fps, *i.e.,* $T_{\mathrm{f}} = 40$ ms. For each sequence, 300 frames are considered. The acquisition delays are taken as constant and equal to the upper bound $T_{\mathrm{a}} = 2$ ms, see [53]. The x265 encoder [54] is configured in low delay mode and with an intra-refresh cycle of one second.

Encoded data packets are embedded in RTP/UDP/IP packets. Wireless transmission of packets is simulated with a period of 1 ms. For this purpose, the 4G trace *A_2018.01.27_10.58.49.csv* from [50] has been considered. Downlink (DL) transmission rates are available with a measurement period of one second. We assume that similar rates are available in the Uplink (UL) direction. Moreover, the transmission rates have been spline interpolated to 1 ms for the transmission simulation. Transmission is assumed to be loss-free thanks to HARQ mechanisms between the transmitter and the base station. The time $T_{\mathrm{c},n}$ spent by packets in the core network is very small compared to other delays and is neglected, as in [24]. For the channel rate prediction performed in Algorithm 2, future channel rates are taken equal to the last observed channel rate. In case of highly dynamic environment, this may lead to inaccurate prediction. More sophisticated predictors may be used, such as [38], [39], [55], see also [40].

Received packets are temporarily stored in the client reception buffer. Decoding starts upon reception of the last packet related to the considered frame. The frame decoding delays are also assumed constant and equal to the upper bound taken as $T_{\mathrm{d}} = 20$ ms, which is consistent with [53]. Decoded frames are stored in a decoded frame buffer before their display, at time $t_n + \Delta_{\mathrm{p}}$ for frame n, where Δ_{p} is the playback delay. This requires a good clock synchronization between the transmitter and the receiver, obtained, e.g., using GPS clocks, or the Network Time Protocol [56]. When a frame is not available in the display buffer, a simple concealment process is realized: lost frames are replaced by the last correctly decoded frame. More

TABLE 3. Performance of MPC for *Park Joy* (640 × 360).

Δ_p (ms)	120	120	120	160	160	160	160	160
τ^* (ms)	20	40	80	10	20	40	60	80
$\overline{PSNR}$	33.39	33.01	29.89	25.21	34.26	34.25	34.15	33.37
$\overline{\lvert\Delta PSNR\rvert}$	0.95	1.25	1.20	1.05	0.92	0.92	0.95	1.24
L	3	2	0	63	1	0	0	0
Δ_p (ms)	200	200	200	200	200	240	240	240
τ^* (ms)	20	40	80	120	160	20	40	80
$\overline{PSNR}$	34.17	34.27	34.25	33.37	29.92	31.32	34.29	34.27
$\overline{\lvert\Delta PSNR\rvert}$	0.93	0.92	0.93	1.24	1.23	0.93	0.92	0.91
L	1	0	0	0	0	16	0	0

TABLE 4. Performance of MPC for *Park Joy* (1280 × 720).

Δ_p (ms)	120	120	120	160	160	160	160	160
τ^* (ms)	20	40	80	10	20	40	60	80
$\overline{PSNR}$	22.32	25.37	24.06	20.77	23.11	25.18	26.85	26.29
$\overline{\lvert\Delta PSNR\rvert}$	0.88	0.82	0.53	0.85	0.73	0.69	0.73	0.77
L	65	20	0	98	44	14	3	0
Δ_p (ms)	200	200	200	200	200	240	240	240
τ^* (ms)	20	40	80	120	160	20	40	80
$\overline{PSNR}$	24.56	26.83	27.02	26.29	24.07	23.43	26.69	27.15
$\overline{\lvert\Delta PSNR\rvert}$	0.70	0.63	0.66	0.77	0.54	0.73	0.64	0.61
L	38	6	1	0	0	54	8	0

sophisticated concealment mechanisms could be considered, e.g. see [37].

For each sequence, the vector of parameters $\mathbf{p}_n$ for frame n of the R-(QP, D) model is estimated iteratively. Apart from the encoder generating the transmitted packets, three additional encoders, operating with time-varying QPs, are used in parallel to provide data to the estimator, i.e., $M = 4$. The choice of $QP_{n,i}$ for frame n and for encoder $i = 1, \ldots, 3$ is performed as follows

$$QP_{n,i} = \begin{cases} QP_{0,i} & \text{if } n = 0 \\ QP_{n-1,i} + \Delta QP_i & \text{if } n\%4 = 1, 2 \\ QP_{n-1,i} - \Delta QP_i & \text{if } n\%4 = 3, 0, \end{cases} \quad (34)$$

where $n\%4$ is the remainder of the division of n by 4. The vector $QP_0 = (24, 36, 40)$ contains the QPs of the first frame, and $\Delta QP = (4, 4, -4)$ is the variation of QP. This choice of QP_0 provides a better model accuracy at low rate (large values of QP). In (28), we have chosen $w_{n,m} = 1/R_{n,m}$ to better balance the importance of measurements obtained at low and high rates. This provides a fit where the relative rate error is approximately constant, whatever the encoding rate. Moreover, we set $\alpha = \lambda_1(\mathbf{X}_n^T\mathbf{W}_n\mathbf{X}_n)/100$, where $\lambda_1(\mathbf{X}_n^T\mathbf{W}_n\mathbf{X}_n)$ is the largest eigenvalue of $\mathbf{X}_n^T\mathbf{W}_n\mathbf{X}_n$. This ensures that $\mathbf{X}_n^T\mathbf{W}_n\mathbf{X}_n + \alpha\mathbf{I}$ is invertible and is sufficient to smooth out the variations of $\widehat{\mathbf{p}}_n$ with n.

In the proposed algorithm, the target playback margin is initially set to $\tau^* = \Delta_p - 2T_f$ when $t \in [0, \Delta_p]$. This ensures a gradual increase of the coding rate to get a smooth video playback at the client. When $t > \Delta_p$, τ^* is set to a constant value depending on the target number τ^*/T_f of frames in the client buffer in permanent regime.

The performance of all algorithms is evaluated using the following metrics (also considered in [26]): $\overline{PSNR}$ is the PSNR averaged over all frames, $\overline{|\Delta PSNR|}$ is the average PSNR absolute variation between two consecutive frames, and L is the number of lost frames.

B. ANALYSIS OF THE PROPOSED MPC ALGORITHM

The proposed MPC algorithm is first evaluated for different values of the initial playback delay Δ_p ranging from 120 ms to 240 ms, and different target playback margins τ^* considering the *Park Joy* sequence. Similar results are obtained for the other sequences. In all simulations, we choose $h = 1$.

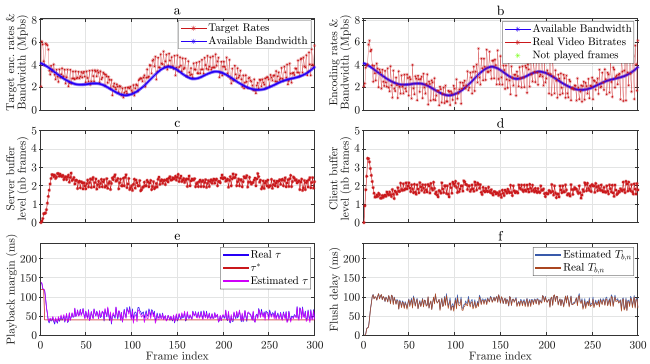

FIGURE 4. MPC for *Park Joy* at 640 × 360 when $\Delta_p = 200$ ms and $\tau^* = 40$ ms: (a) evolution of the target and transmission rates, (b) actual encoding rates, (c) transmission buffer level, (d) client buffer level, (e) actual and estimated value of τ, (f) actual and estimated values of $T_{b,n}$.

Tables 3 and 4 summarize the results at resolutions 640 × 360 and 1280 × 720 respectively. As expected, a larger playback delay ($\Delta_p = 200$ ms or $\Delta_p = 240$ ms) leads to fewer losses, since transmission rate variations are better handled. Similarly, large values of τ^* provide also better performance. For example, when $\Delta_p = 200$ ms and $\tau^* = 160$ ms, the control is performed to provide four frames in the client buffer. In such a regime, the MPC algorithm adjusts the encoding rate without causing any frame loss. Nevertheless, large values of τ^* lead to conservative encoding rate selections, which decreases the average PSNR of the decoded video.

When τ^* is too small, frames may be lost. The MPC algorithm becomes less conservative and tries to better exploit the available transmission rate by encoding frames with a higher rate. Nevertheless, the inaccuracy of the R-(QP, D) model may lead to encoded frames at a rate higher than the target rate R_n^*, leading to increased transmission delays. Similar effects are obtained when the transmission rate has been overestimated. The playback margin τ^* helps mitigate these discrepancies. For a video at 1280 × 720, $\tau^* = 80$ ms provides good results even with a very small end-to-end playback delay of $\Delta_p = 120$ ms.

Fig. 4(a) shows the evolution of the target encoding rate for the *Park Joy* sequence at resolution 640× 360 when $\tau^* = 40$ ms, along with the channel capacity. Fig. 4(b) shows the evolution of the actual encoding rate and the channel capacity. One observes in Fig. 4(a) that the target encoding rate is slightly higher in average than the channel capacity. The MPC

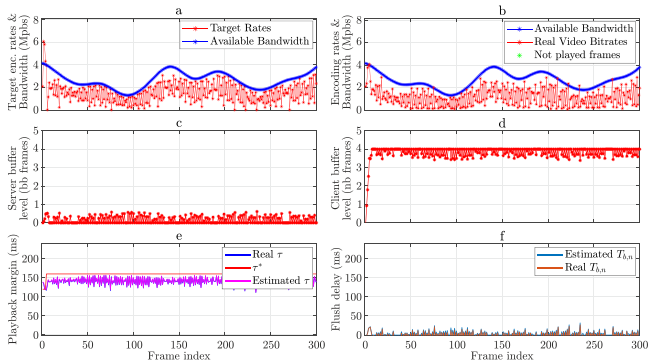

FIGURE 5. MPC for *Park Joy* at 640 × 360 when $\Delta_p = 200$ ms and $\tau^* = 160$ ms: (a) evolution of the target and transmission rates, (b) actual encoding rates, (c) transmission buffer level, (d) client buffer level, (e) actual and estimated value of τ, (f) actual and estimated values of $T_{b,n}$.

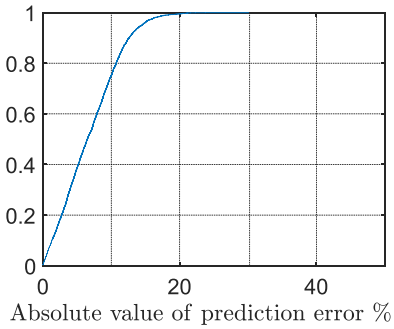

FIGURE 6. CDF of the absolute value of the relative error rate obtained with the R-(QP, D) model considering ten independent transmission trials of the video sequence *DaylightRoad2* at 640 × 360 using the proposed frame-level MPC rate control algorithm.

approach is able to compensate for the bias introduced by the R-(D,QP) model in the selection of the QP matching the rate target, as seen in Fig. 4(b) where the actual encoding rate oscillates around the channel capacity. Figures 5(a) and 5(b) show target and actual encoding rates when $\tau^* = 160$ ms. Imposing a large τ^* leads to a conservative use of the channel capacity. When $\tau^* = 40$ ms, the target encoding rates selected by the MPC approach are close to the available transmission rates. Conversely, when $\tau^* = 160$ ms, the selected target rates are always less than the transmission rate to ensure the satisfaction of the playback margin target. Smaller values of τ^* lead to a better exploitation of the available channel capacity.

Fig. 4(d) shows that the client buffer contains only 1.5 frames when $\tau^* = 40$ ms. Conversely, Fig. 5(d) shows that the client buffer contains almost always about 4 frames when $\tau^* = 160$ ms. Setting large τ^* provides a large margin to react to sudden drops of the transmission rate. The value of τ^* determines the trade-off reached between bandwidth exploitation and protection against sudden drops of the transmission rate.

Fig. 4(e) shows the evolution of the estimated and actual values of τ, as well as the estimated and actual values of the flushing delay of the transmission buffer $T_{b,n}$, when $\tau^* = 40$ ms. Fig. 5(e) shows similar results when $\tau^* = 160$ ms. The estimates of τ and $T_{b,n}$ are quite accurate when using the last measure of the transmission rate as observed in Figs 4(f) and 5(f). The proposed algorithm has difficulties to maintain the actual value of τ at the level of the target playback margin τ^* when τ^* is too small or too close to Δ_p due to coded packets stored in the transmission buffer and packets passing through the network.

C. PERFORMANCE OF THE R-(QP, D) MODEL

This section evaluates the prediction error of the R-(QP, D) model when its parameters are estimated recursively as described in Section IV. The prediction performance is evaluated using the relative rate error $E_n = 100(R_{m,n} - R_n)/R_n$, where R_n and $R_{m,n}$ are respectively the observed size of the n-th encoded frame and the size as predicted by the R-(QP, D) model for the selected value of QP. Fig. 6 shows the Cumulative Distribution Function (CDF) of the absolute value of the relative rate error observed for ten transmission episodes of the video sequence *DaylightRoad2* at resolution 640 × 360 when the MPC algorithm is used for the rate control. For more than 75 % of the frames, the absolute value of the relative error E_n is less than 10 % . This confirms the very good performance of the R-(QP, D) model.

D. PERFORMANCE COMPARISON WITH REFERENCE ALGORITHMS

The proposed rate control algorithm is compared to Festive [11], Panda [12], BOLA [13], and BBA [14], adapted to get frame-level, server-driven encoding rate controllers. The R-(QP, D) model is used by all these algorithms to obtain the target value QP_n^* from the selected target encoding rate R_n^*.

Results are considered for ten independent transmission episodes for each video sequence. Each episode considers a different starting time instant in the bandwidth trace. The playback delay is set to $\Delta_p = 200$ ms for all algorithms. For the MPC algorithm, $\tau^* = 50$ ms when the frame resolution is 640 × 360 and $\tau^* = 80$ ms when it is 1280 × 720.

Tables 5 and 6 summarize the results obtained with each rate control algorithm and for frame sizes of 640 × 360 and 1280 × 720. The proposed MPC algorithm provides the best performance in terms of average PSNR and lost frames for all sequences. The largest frame loss for the proposed algorithm is obtained with *Park Joy* at resolution 1280 × 720, where 5 frames are lost among 3000 transmitted ones. This is due to a reduced accuracy of the R-(QP, D) model when the characteristics of the video sequence vary quickly, which makes the iterative estimation of the model parameters more challenging. The downside of the proposed approach is a greater variability, albeit still acceptable, of the PSNR of encoded frames. Panda and Festive, which are both bandwidth-based algorithms, yield the smallest PSNR variability. In contrast, the other algorithms try to stabilize the buffer levels, which results in oscillations of the encoding rates and of the PSNR.

TABLE 5. Performance of the Proposed MPC Algorithm Compared to Festive [11], Panda [12], BOLA [13], and BBA [14] With a Frame Size of 640 × 360.

Sequence	Method	L	$\overline{PSNR}$	$\overline{\lvert\Delta PSNR\rvert}$	$\overline{SSIM}$	$\overline{VMAF}$
CrowdRun	MPC	**0**	**33.35**	0.55	**0.928**	**96.85**
	BBA	1	33.25	0.35	**0.928**	96.80
	Festive	6	29.92	0.17	0.893	91.83
	Panda	**0**	30.80	**0.16**	0.891	91.45
	BOLA	**0**	32.99	1.25	0.926	95.87
ParkJoy	MPC	**0**	**34.27**	0.92	**0.931**	**96.54**
	BBA	5	32.97	0.59	0.891	88.62
	Festive	**0**	31.86	**0.27**	0.900	92.58
	Panda	**0**	31.74	**0.27**	0.898	91.49
	BOLA	**0**	33.84	1.27	0.930	95.97
Touch-	MPC	**1**	**44.58**	0.61	**0.983**	**99.40**
DownPass	BBA	95	27.98	0.64	0.782	23.60
	Festive	36	37.99	0.32	0.903	69.72
	Panda	24	40.20	**0.27**	0.929	79.08
	BOLA	0	44.38	0.92	0.982	99.27
Daylight-	MPC	**0**	**44.32**	0.33	**0.986**	**99.75**
Road2	BBA	**0**	44.31	0.19	**0.986**	99.74
	Festive	**0**	42.87	**0.11**	0.982	99.70
	Panda	**0**	42.72	**0.11**	0.982	99.67
	BOLA	**0**	44.10	0.88	0.986	99.69
Kristen-	MPC	**0**	**48.28**	0.08	**0.992**	**98.38**
and Sara	BBA	**0**	48.27	0.07	**0.992**	**98.38**
	Festive	**0**	47.64	**0.09**	0.991	98.30
	Panda	**0**	47.58	**0.09**	0.991	98.30
	BOLA	**0**	48.20	0.10	0.992	98.36

TABLE 6. Performance of the Proposed MPC Algorithm Compared to Festive [11], Panda [12], BOLA [13], and BBA [14] With a Frame Size of 1280 × 720.

Sequence	Method	L	$\overline{PSNR}$	$\overline{\lvert\Delta PSNR\rvert}$	$\overline{SSIM}$	$\overline{VMAF}$
CrowdRun	MPC	**0**	**28.26**	0.30	**0.821**	**79.08**
	BBA	**0**	28.25	0.18	**0.821**	79.04
	Festive	**0**	26.90	**0.13**	0.781	69.97
	Panda	**0**	26.86	**0.13**	0.780	69.74
	BOLA	**0**	28.15	0.46	0.820	78.24
ParkJoy	MPC	5	**27.03**	0.65	**0.806**	**76.25**
	BBA	20	26.81	0.62	0.800	74.19
	Festive	7	24.93	**0.42**	0.737	60.10
	Panda	5	24.90	**0.42**	0.735	60.08
	BOLA	**1**	26.28	0.71	0.787	71.49
Touch-	MPC	**0**	**40.14**	0.46	**0.953**	**95.33**
DownPass	BBA	34	37.26	0.40	0.915	16.45
	Festive	6	38.73	0.33	0.936	77.34
	Panda	**0**	38.74	**0.32**	0.937	67.69
	BOLA	**0**	39.74	0.50	0.949	64.55
Daylight-	MPC	**0**	**40.10**	0.26	**0.967**	**76.25**
Road2	BBA	**0**	40.09	0.14	**0.967**	74.19
	Festive	**0**	38.87	**0.08**	0.961	60.10
	Panda	**0**	38.79	**0.08**	0.960	60.08
	BOLA	**0**	39.93	0.55	0.964	71.49
Kristen-	MPC	**0**	**44.18**	0.24	**0.981**	**97.75**
and Sara	BBA	**0**	44.17	0.13	**0.981**	**97.75**
	Festive	**0**	43.43	**0.09**	0.979	97.42
	Panda	**0**	43.38	**0.09**	0.979	97.39
	BOLA	**0**	44.05	0.30	0.980	97.64

For all sequences, this variation is less than 1 dB (and often much smaller), which is usually unnoticeable by observers.

Compared to the proposed algorithm, BOLA achieves a slightly lower average PSNR quality and leads to more lost frames. When BOLA achieves 0 lost frames, it comes at the cost of a lower average PSNR and larger variations. The BBA algorithm has the worst performance as it tends to be too aggressive by selecting a high encoding rate when the buffer level allows it. This causes a large number of frame losses, especially when the R-(QP, D) model is less accurate. The PSNR variations when using BBA is usually smaller than those observed with BOLA and the proposed approach.

Tables 5 and 6 also show the performance of the rate adaptation algorithms in terms of average SSIM and VMAF. The MPC algorithm also provides the best performance considering these metrics for all video sequences.

Fig. 7 (left) shows the evolution with time of the transmission rate, and of the target and actual encoding rates for all algorithms in the second transmission episode of *DaylightRoad2* at resolution 640 × 360, when $\mathbf{\Delta}_{\mathrm{p}} = 200$ ms and $\tau^* = 50$ ms. Festive and Panda have close behavior in the selection of the target encoding rate. These two rate-based approaches are conservative and select a target encoding rate lower than the channel transmission rate. BOLA leads to large rate oscillations. This explains the fact that BOLA has the largest average PSNR variations. The proposed approach and BBA have an overall similar behavior, even if the proposed approach follows better the variations of the transmission channel compared to BBA. In addition, BBA leads to slightly larger oscillations of the PSNR. This has been verified with the other transmission episodes and the other video sequences. Fig. 7 also illustrates the accuracy of the R(D, QP) model: in most cases, the actual encoding rate is close to the target encoding rate. This shows that the QP values determined by the model are adequate.

Fig. 7 (middle) shows the evolution of the transmission and client buffer levels for the previous algorithms. Festive and Panda, the two rate-based algorithms, keep the client buffer level high, as they select an encoding rate preventing an empty client buffer. Conversely, the buffer level with BOLA oscillates as the selected target rate is continuously changing. In addition, as BOLA is less conservative, the buffer level has a lower value than with Festive or Panda. The buffer level of BBA oscillates around the same value as that obtained by the proposed MPC approach, but the latter is much more stable.

Fig 7 (right) illustrates the evolution of the PSNRs obtained at the receiver with the previous algorithms. The PSNR variations follow in a smoothed way those of the encoding rate in Fig. 7 (left). The oscillating behavior of the rate for BOLA is also observed in the PSNR. Small amplitude oscillations (0.5–0.7 dB) are also seen for the proposed MPC approach compared to BBA, Festive, and Panda. This is mainly due to the closer adaptation of the encoding rate to the channel rate of the MPC approach. This leads to an increased average PSNR, but also to more frequent corrections of the encoding rate to meet the target playback margin.

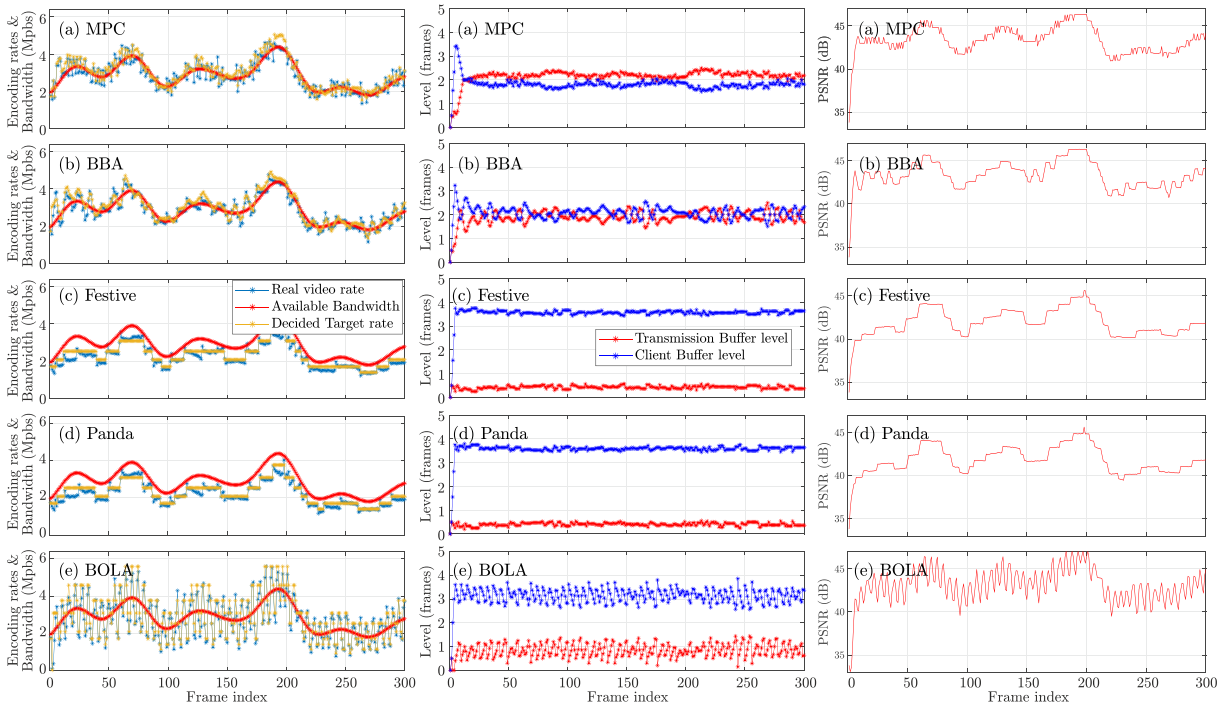

FIGURE 7. Evolution of the transmission rate, the selected target rate, and the actual encoding rate (left) and of the transmission and client buffers (right) for *DaylightRoad2* at resolution 640 × 360, when $\Delta_p = 200$ ms and $\tau^* = 50$ ms, considering the proposed MPC, BBA [14], Festive [11], Panda [12], BOLA [13] algorithms.

VII. CONCLUSION

This paper presents a new MPC algorithm for frame-level encoding rate adaptation targeting ULLVCD for video acquired within a mobile device. It exploits the transmission buffer level and an estimate of the wireless transmission rate to determine the target encoding rate of each frame. The choice of the QP for each frame is performed via an R-(QP, D) model, able to predict the size of the current encoded frame as a function of its QP and the distortion of the reference frame.

Considering a video encoding and delivery application with a G2G latency less than 200 ms, the proposed approach outperforms four reference algorithms, Festive [11], Panda [12], BOLA [13], and BBA [14], in terms of average PSNR and frame losses.

The accuracy of the estimate of the transmitter buffer level and of the transmission rate are instrumental in ULLVCD. This type of information is provided at a high frequency by tools such as MobileInsight [57] in the mobile device where encoding rate control has to be performed.

This work may be improved in several direction. Databases containing geo-located measurements of the channel quality may be combined with Kalman-based prediction of the transmitter position [38], [39] to get long-term estimates of the wireless channel capacity so as to improve the prediction accuracy of the time required to flush the encoded frames from the transmission buffer. This information may also be used to adjust dynamically the target playback margin τ^* and the end-to-end playback delay Δ_p by small variations of the frame rate at receiver [58], in order to maximize the QoE for a given latency constraint. The prediction accuracy of the R-(QP, D) model may also be improved by detecting changes in scene illumination and by accounting for the motion activity of the video sequence. The proposed approach would also benefit from an alternative, lower-complexity R-(QP, D) model which would not require several encoding trials for each frame in order to adapt the model parameters. Techniques exploiting the temporal and spatial information such as [59] for learned video compression may be interesting to facilitate the embedding of the proposed solution on a lightweight platform, such as a drone.

REFERENCES

[1] Y. Sato, S. Kashihara, and T. Ogishi, "Robust video transmission system using 5G/4G networks for remote driving," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2022, pp. 616–622.
[2] D. Mejías, Z. Fernández, R. Viola, A. Aramburu, I. Lopez, and A. Diaz, "Towards railways remote driving: Analysis of video streaming latency and adaptive rate control," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit*, 2024, pp. 1090–1095.
[3] J. Y. C. Chen, E. C. Haas, and M. J. Barnes, "Human performance issues and user interface design for teleoperated robots," *IEEE Trans. Syst. Man Cybern. Part C*, vol. 37, no. 6, pp. 1231–1245, Nov. 2007.
[4] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.
[5] H. Ullah, N. Gopalakrishnan Nair, A. Moore, C. Nugent, P. Muschamp, and M. Cuevas, "5G communication: An overview of Vehicle-to-Everything, drones, and healthcare use-cases," *IEEE Access*, vol. 7, pp. 37251–37268, 2019.

[6] M. Hofbauer, C. B. Kuhn, M. Khlifi, G. Petrovic, and E. Steinbach, "Traffic-aware multi-view video stream adaptation for teleoperated driving," in *Proc. IEEE Veh. Technol. Conf.*, 2022, pp. 1–7.

[7] M. Yamaguchi, M. Yamakawa, S. Mochizuki, K. Imamura, T. Kanamoto, and T. Matsumura, "Ultra-low-latency video coding with reduced frame memory structure for 4K/8K high-resolution video," in *Proc. 2023 IEEE 12th Glob. Conf. Consum. Electron.*, 2023, pp. 852–853.

[8] N. Francisco, O. Baumann, J. L. Tanou, and R. Fliam, "Ultra-low latency video delivery over WebRTC data channels," in *Proc. 3rd Mile-High Video Conf.*, New York, NY, USA, 2024, pp. 88–89.

[9] "Ultra-low latency delivery over IP," Tech. Rep. [Online]. Available: https://zixi.wpenginepowered.com/wp-content/uploads/2020/06/Zixi-Ultra-Low-Latency-Delivery.pdf&ved=2ahUKEwjrp8eZqtCOAxX0K_sDHb72HxAQFnoECD8QAQ&usg=AOvVaw3HlErVpZ_2JtdHnjIEqmDT

[10] 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 Tut.*, vol. 21, no. 1, pp. 562–585, Firstquarter 2019.

[11] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive," in *Proc. 8th Int. Conf. Emerg. Netw. Experiments Technol.*, 2012, pp. 97–108.

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

[13] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1698–1711, Aug. 2020.

[14] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proc. ACM Conf. SIGCOMM*, 2014, pp. 187–198.

[15] P. K. Yadav, A. Bentaleb, M. Lim, J. Huang, W. T. Ooi, and R. Zimmermann, "Playing chunk-transferred dash segments at low latency with qlive," in *Proc. 12th ACM Multimedia Syst. Conf.*, New York, NY, USA, 2021, pp. 51–64.

[16] I. M. Ozcelik and C. Ersoy, "ALVS: Adaptive live video streaming using deep reinforcement learning," *J. Netw. Comput. Appl.*, vol. 205, 2022, Art. no. 103451.

[17] R. Peck, J. Cenzano, X. Li, and Y. Reznik, "Towards mass deployment of CMAF," in *Proc. NAB Broadcast Eng. Inf. Technol. Conf.*, 2019, pp. 1–6.

[18] T. Feng, H. Sun, Q. Qi, J. Wang, and J. Liao, "Vabis: Video adaptation bitrate system for time-critical live streaming," *IEEE Trans. Multimedia*, vol. 22, pp. 2963–2976, 2020.

[19] A. Aloman, A. Ispas, P. Ciotirnae, R. Sanchez-Iborra, and M.-D. Cano, "Performance evaluation of video streaming using MPEG DASH, RTSP, and RTMP in mobile networks," in *Proc. 8th IFIP Wireless Mobile Netw. Conf.*, 2015, pp. 144–151.

[20] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)" RFC 2326, Apr. 1998.

[21] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 1889, Jan. 1996.

[22] O. E. Marai and T. Taleb, "Smooth and low latency video streaming for autonomous cars during handover," *IEEE Netw.*, vol. 34, no. 6, pp. 302–309, Nov./Dec. 2020.

[23] I. Khan, T. X. Tran, M. Hiltunen, T. Karagioules, and D. Koutsonikolas, "An experimental study of low-latency video streaming over 5G," in *Proc. IEEE Int. Mediterranean Conf. Commun. Netw.*, 2024, pp. 383–388.

[24] C. Bachhuber, E. Steinbach, M. Freundl, and M. Reisslein, "On the minimization of glass-to-glass and glass-to-algorithm delay in video communication," *IEEE Trans. Multimedia*, vol. 20, pp. 238–252, 2018.

[25] B. Bruno, B. Vizzotto, M. Zatt, S. S. Bampi, and J. Henkel, "A model predictive controller for frame-level rate control in multiview video coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2012, pp. 485–490.

[26] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *Proc. ACM Conf. Special Int. Group Data Commun.*, New York, NY, USA, 2015, pp. 325–338.

[27] M. Aklouf, M. Leny, M. Kieffer, and F. Dufaux, "Interframe-dependent Rate-QP-Distortion model for video coding and transmission," in *Proc. 2021 IEEE Int. Conf. Image Process.*, 2021, pp. 2019–2023.

[28] S. Ubik and J. Pospíšilík, "Video camera latency analysis and measurement," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 140–147, Jan. 2021.

[29] 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.

[30] Y. Huang, J. Xu, C. Zhu, L. Song, and W. Zhang, "Precise encoding complexity control for versatile video coding," *IEEE Trans. Broadcast.*, vol. 69, no. 1, pp. 33–48, Mar. 2023.

[31] Y. Zhao, C. Zhu, J. Xu, G. Lu, L. Song, and S. Ma, "An efficient and flexible complexity control method for versatile video coding," *IEEE Trans. Broadcast.*, vol. 71, no. 1, pp. 96–110, Mar. 2025.

[32] S. J. Udoh and V. M. Srivastava, "Analytical modeling of radio network performance for 5 G (non-standalone) and it's network connectivity," *J. Commun.*, vol. 15, no. 12, pp. 886–895, 2020.

[33] L. Kundu, G. Xiong, and J. Cho, "Physical uplink control channel design for 5 G new radio," in *Proc. IEEE 5G World Forum*, 2018, pp. 233–238.

[34] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[35] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.

[36] A. R. Abdellah, O. A. Mahmood, R. Kirichek, A. Paramonov, and A. Koucheryavy, "Machine learning algorithm for delay prediction in IoT and tactile internet," *Future Internet*, vol. 13, no. 12, 2021, Art. no. 304.

[37] M. Kazemi, M. Ghanbari, and S. Shirmohammadi, "A review of temporal video error concealment techniques and their suitability for HEVC and VVC," *Multimedia Tools Appl.*, vol. 80, no. 8, pp. 12685–12730, 2021.

[38] J. Hao, R. Zimmermann, and H. Ma, "Gtube: Geo-predictive video streaming over HTTP in mobile environments," in *Proc. 5th ACM Multimedia Syst. Conf.*, 2014, pp. 259–270.

[39] D. Salcedo, J. Guerrero, and C. D. Guerrero, "Overhead in available bandwidth estimation tools: Evaluation and analysis," *Int. J. Commun. Netw. Inf. Secur.*, vol. 9, no. 3, pp. 393–404, 2017.

[40] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, Fourthquarter 2019.

[41] L. Wang, S. Hong, and K. Panusopone, "Gradual decoding refresh for versatile video coding," in *Proc. IEEE Int. Conf. Image Process.*, 2021, pp. 3448–3452.

[42] H. Esmaeeli and M. Rezaei, "Proportional-integral-derivative-based intra rate controller for low-delay applications of high efficiency video coding screen content coding," *J. Electron. Imag.*, vol. 31, no. 2, 2022, Art. no. 023029.

[43] Y. Chen, M. Wang, S. Wang, Z. Ni, and S. Kwong, "A CTU-level screen content rate control for low-delay versatile video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 5227–5241, Sep. 2023.

[44] H. Wang, X. Zhang, H. Chen, Y. Xu, and Z. Ma, "Inferring end-to-end latency in live videos," *IEEE Trans. Broadcast.*, vol. 68, no. 2, pp. 517–529, Jun. 2022.

[45] M. Zhang, W. Zhou, H. Wei, X. Zhou, and Z. Duan, "Frame level rate control algorithm based on GOP level quality dependency for low-delay hierarchical video coding," *Signal Process., Image Commun.*, vol. 88, 2020, Art. no. 115964.

[46] I. Marzuki, J. Lee, and D. Sim, "Optimal CTU-level rate control model for HEVC based on deep convolutional features," *IEEE Access*, vol. 8, pp. 165670–165682, 2020.

[47] Z. Li et al., "An optimized algorithm at CTU-level for rate control," in *Proc. Int. Conf. Bioinf. Biomed. Technol.*, 2021, pp. 33–40.

[48] H. Choi, J. Yoo, J. Nam, D. Sim, and I. V. Bajić, "Pixel-wise unified rate-quantization model for multi-level rate control," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1112–1123, Dec. 2013.

[49] E. Wlater and L. Pronzato, *Identification of Parametric Models From Experimental Data*, Berlin, Germany: Springer, 1997.

[50] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond throughput: A 4G LTE dataset with channel and context metrics," in *Proc. 9th ACM Multimedia Syst. Conf.*, 2018, pp. 460–465.

[51] T. Suzuki, "JVET-D1002: Work plan for assessment of test material," in *Proc. JVET 4th Meeting*, Chengdu, China, 2016.
[52] FFmpeg Developers, "ffmpeg tool (build: Ffmpeg-20180716-8aa6d9a-win64-static)," 2019. [Online]. Available: http://ffmpeg.org/
[53] iWave, "Ultra-low latency video transmitting and receiving system." [Online]. Available: https://www.iwavesystems.com/news/ultra-low-latency-video-transmitting-and-receiving-system/&ved=2ahUKEwiMwI2-r9COAxXQVKQEHULMIOoQFnoECBkQAQ&usg=AOvVaw29vILh5r842FMF3NmHw815
[54] MulticoreWare, "x265 software documentation," 2020. [Online]. Available: https://x265.readthedocs.io/en/master/
[55] J. Lee et al., "PERCEIVE: Deep learning-based cellular uplink prediction using real-time scheduling patterns," in *Proc. 18th Int. Conf. Mobile Syst., Appl., Serv.*, New York, NY, USA, 2020, pp. 377–390.
[56] D. L. Mills, "Network time protocol (Version 3) specification, implementation and analysis," RFC 1305, Mar. 1992.
[57] L. Yuanjie, P. Chunyi, and L. Songwu, "Mobileinsight, a fine-grained mobile network analytics inside the smartphones," 2021. [Online]. Available: http://www.mobileinsight.net/
[58] G. Zhang and J. YB Lee, "LAPAS: Latency-aware playback-adaptive streaming," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2019, pp. 1–6.
[59] J. Li, B. Li, and Y. Lu, "Hybrid spatial-temporal entropy modelling for neural video compression," in *Proc. 30th ACM Int. Conf. Multimedia*, New York, NY, USA, 2022, pp. 1503–1511.

MOURAD AKLOUF received the M.Sc. degree in telecommunication from Télécom Paris, in 2018, and the Ph.D. degree in image and signal processing from CentraleSupélec, Université Paris-Saclay, France, in 2022. His PhD thesis focused on low latency video streaming, adaptive video coding, and video bitrate control. He is currently an OTT/IPTV Video Architect with several media companies in France. He is interested in various topics related to the video industry, such as the optimization of video codecs, OTT, IPTV, content delivery network, AI applications to video streaming and compression.

FRÉDÉRIC DUFAUX (Fellow, IEEE) received the M.Sc. degree in physics and the Ph.D. degree in electrical engineering from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 1990 and 1994 respectively. He is currently a CNRS Research Director with Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes (L2S, UMR 8506), where he is the Head of the Telecom and Networking Research Hub. He is the author or coauthor of three books, more than 250 research publications, and more than 25 patents issued or pending. His research interests include image and video coding, 3-D video, high dynamic range imaging, visual quality assessment, video surveillance, privacy protection, image and video analysis, multimedia content search and retrieval, and video transmission over wireless networks. He is in the World's Top 2% Scientists list from Stanford University. Frederic was the Vice General Chair of ICIP 2014, General Chair of MMSP 2018, Technical Program Co-Chair of ICIP 2019 and ICIP 2021, Chair of the IEEE SPS Multimedia Signal Processing (MMSP) Technical Committee in 2018 and 2019, respectively, and the Chair of the Steering Committee of ICME in 2022 and 2023, respectively. He is the Technical Program Co-Chair of ICIP 2025 and MMSP 2025, and the General Chair of ICME 2026. Since 2025, he has been the IEEE SPS Vice President Technical Directions, and a member of the IEEE SPS Board of Governors and Executive Committee. He was also a Founding Member and the Chair of the EURASIP Technical Area Committee on visual information processing from 2015 to 2021. He was the Editor-in-Chief of *Signal Processing: Image Communication* from 2010 until 2019. Since 2021, he has been the Specialty Chief Editor of the section on Image Processing in the journal *Frontiers in Signal Processing*. He has been the Executive Board of Systematic Paris-Region since 2019, European competitiveness cluster which brings together and drives an ecosystem of excellence in digital technologies and DeepTech.

MICHEL KIEFFER (Senior Member, IEEE) received the Ph.D. degree in control theory from the University of Paris XI, Orsay, France, in 1999. From 2009 to 2016, he has been a part-time Invited Professor with the Laboratoire Traitement et Communication de l'Information, Télécom ParisTech, Paris, France. He was a Junior Member of the Institut Universitaire de France from 2011 to 2016. He is currently a Full Professor of signal processing for communications with Université Paris-Saclay and a Researcher with the Laboratoire des Signaux et Systèmes (L2S), Gif-sur-Yvette, France. He is the coauthor of more than 230 contributions in journals, conference proceedings, or books. He is one of the coauthors of the books *Applied Interval Analysis* (Springer-Verlag, 2001) and *Joint Source-Channel Decoding: A Cross-Layer Perspective With Applications in Video Broadcasting* (Academic, 2009). His research interests include signal processing for multimedia, communications, and networking, source coding, network coding, joint source-channel coding and decoding techniques, and joint source-network coding. Applications are mainly in the reliable delivery of multimedia contents over wireless channels. He is also interested in guaranteed and robust parameter and state bounding for systems described by nonlinear models in a bounded-error context. He has been an Associate Editor for *Signal Processing,* since 2008 and IEEE TRANSACTIONS ON COMMUNICATIONS from 2012 to 2016.

MARC LÉNY received the Engineering degree in telecommunications from Institut Mines-Telecom (IMT), Paris, France, and the Master of Engineering degree in electronic systems from Dublin City University, Dublin, Ireland, and the Ph.D. degree in 2010, led jointly with IMT and Thales Communications, focusing on video analysis in the compressed domain. In 2012, he joined Ektacom, where he is in charge of developing the R&D activities to prepare the new products to the upcoming multimedia standards. He has coordinated European and French collaborative projects and handle work-packages and use-cases. His research interests include AI for video analysis, compression or transmission, custom designed and out-of-the-box thinking for demanding customers.