IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

Adaptive Cloud-Based Extended Reality: Modeling and Optimization

MIKHAIL LIUBOGOSHCHEV[1,2], KAMILA RAGIMOVA[1,2], ANDREY LYAKHOV[1] (Member, IEEE), SIYU TANG[3], EVGENY KHOROV[1] (Senior Member, IEEE),

[1]Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Moscow, Russia (e-mail: {liubogoshchev, ragimova, lyakhov, khorov}@wireless.iitp.ru)
[2]Moscow Institute of Physics and Technology, Moscow, Russia
[3]Huawei Munich Research Center, Munich, Germany (e-mail: siyutang@huawei.com)

Corresponding author: Evgeny Khorov (e-mail: khorov@wireless.iitp.ru).

The research was done at IITP RAS and supported by Moscow and Munich Research Centers of Huawei.

ABSTRACT

Extended Reality (XR) — which includes Virtual Reality and Augmented Reality — promises to bring the virtual and telepresence experience to another level. Unfortunately, solutions leveraging these technologies require special high-performance computing platforms that degrade the cost-benefit balance. Moving processing to the cloud solves this problem but imposes strict requirements on data transmission reliability, bandwidth, and delays. The satisfaction of these requirements becomes an extremely challenging problem in the presence of other types of delay-sensitive traffic, such as remote control, industrial automation, or the control commands of the Cloud XR application itself. This article studies the joint service of the adaptive Cloud XR traffic with other high-priority delay-sensitive traffics. The paper develops an analytical model of the considered communication system. The model represents the system as a discrete state Markov chain and estimates the quality of experience for Cloud XR users in various scenarios. Using the model, the paper estimates the network capacity for the Cloud XR traffic and optimizes the bitrate adaptation function of the Cloud XR video streaming application. The goal of the optimization is to improve the visual quality of the virtual environment observed by the users, subject to the constrained probability of image impairments due to excessive delivery delays. Numerical results demonstrate the high accuracy of the developed model and the benefits provided by the optimization.

INDEX TERMS Cloud XR, heterogeneous traffic, high-priority traffic, real-time adaptive video, virtual reality, quality of experience, queueing systems, analytical models, Markov chain

I. INTRODUCTION

Extended Reality (XR), which includes Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), is one of the key technologies enabling virtual and telepresence. Numerous studies and emerging technological products reveal that XR technologies can be applied in various fields. For example, in education, highly immersive XR applications can improve the attention and interest of the students [1]. In medicine, XR applications can be used for clinical protocol testing and educational training [2]. In engineering, architecture, and geo-informational sciences, XR technologies simplify modeling, visualization, and analysis of large-scale complex structures [3–6].

However, the solutions leveraging XR non-tethered to a workstation require integrating special high-performance computing platforms into battery-powered XR-headsets. This simultaneously constrains the achievable visual quality, reduces the battery life, and increases the cost of the headsets [7]. A recent paradigm of Cloud XR moves most of the processing to the cloud and changes the system architecture as follows.

In Cloud XR [8], the headset does not render the virtual scene by itself, so it does not require expensive and power-consuming hardware. Instead, the headset captures the user's actions and sends the data to a remote server. The server renders the virtual XR scene according to the received data, encodes it into a video stream, and sends

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

it back to the headset that shows the video to the user. Moving processing to the cloud makes headsets very cheap and reduces their weight and power consumption [9], but imposes strict requirements on data transmission reliability, bandwidth, and delays between the remote cloud server and the end-users [10]. Such an architecture might look fantastic even a decade ago, but today with multi-gigabit wired and wireless links, the architecture seems feasible [11] and is evaluated and deployed around the world [12–14].

Typically, XR applications are interactive. To provide an immersive experience, Cloud XR applications require minimal feedback delay and high image quality [15, 16]. Since the content is generated on-the-fly according to the actions of the user, each generated video frame shall be delivered to the headset with a limited delay. However, in the networks, video traffic may have lower priority than the other delay-sensitive traffic: remote control, gaming, industrial automation. In addition to interference from these traffic types, video traffic transmissions can be affected by the control traffic of the XR application itself. Therefore, the amount of network resources available to the XR video stream can fluctuate with time, and some video frames may be delivered longer than others. To prevent playback interruptions and to ensure the maximal image quality, XR applications shall adaptively select the quality of the video stream in real-time [17]. However, the tight latency requirements and sporadic interference from higher-priority flows render it challenging for the adaptation algorithm to strike the right balance between resiliency and image quality.

The optimization of the Quality of Experience (QoE) for XR and Cloud XR applications received much attention from both researchers and industry. XR videos are panoramic, so novel approaches to efficiently represent and compress it were developed [18] and standardized [19]. Next, to increase the image refresh rate, different approaches to motion prediction and proactive video rendering and transmission were devised [20, 21].

Other important aspects of Cloud XR optimization lie in finding the right balance between the computations performed by the headset and the computations performed by the cloud or Mobile Edge Computing server [22] and optimizing the energy-efficiency of the cloud [23–25].

Finally, many papers propose different strategies to jointly optimize XR scene rendering, caching the proactively generated video frames and their transmission by reserving the network resources [26–28]. However, typically authors do not consider the structure of the XR video flows and the organization of the XR presentation at the headsets. Instead, they reduce the video stream to a series of requests that shall be processed under a fixed delay constraint. Such a traffic pattern is more relevant for WebXR services, e.g., [29–31]. Also, the typical assumption is that the throughput provided by the network to the XR flows is constant, or a sufficient amount of network resources can be reserved when needed. However, no interference from the other traffics in the network is considered. That is why such papers do not focus on the XR video quality adaptation.

In contrast to these papers, we look at the Cloud XR QoE optimization from a different perspective. We model the transmission of the XR data through the network and take into account the variation in its delivery rate caused by random interference from other high-priority traffics.

In the paper, we study the joint service of adaptive Cloud Extended Reality application traffic with other high-priority delay-sensitive traffics. *The contribution of the paper is as follows.*

First, to the best of our knowledge, we are first to design a mathematical model of an adaptive real-time Cloud XR application that allows evaluating QoE for XR in wired and wireless networks. For that, we take into account the following peculiarities of the traffic generated by XR applications often left out of consideration in the literature.

1) We consider a realistic client-side XR application design that employs a small jitter-buffer to smoothen the fluctuations in video frame delivery.
2) We consider that the XR traffic consists of two traffic types: high-priority control traffic and real-time adaptive XR video.
3) We take into account that the priority of XR video flows may be lower than that of other delay-sensitive traffic types.

We consider the average video bitrate and stalling probability as objective video QoE metrics because they provide a good trade-off between modeling accuracy and estimation complexity. Also, they are often considered in the literature [32]. For control traffic, we take into account the mean command delivery delay.

Second, we use the developed model to estimate the capacity of a communication system for XR video flow. We define the capacity as the maximal average XR video bitrate for which its delay requirements can be met with a pre-defined probability. Finally, we use the model to find the XR video bitrate adaptation function that maximizes the capacity.

The rest of the paper is organized as follows. In Section II, we describe the Cloud XR system and introduce the problem statement. Section III reviews the relevant literature. Section IV describes the joint service model, how it can be used to estimate the network capacity for the XR video stream and to optimize the XR bitrate adaptation function. In Section V, we present and discuss the obtained numerical results. Finally, Section VI concludes the paper.

II. SYSTEM DESCRIPTION AND PROBLEM STATEMENT

A. SYSTEM DESCRIPTION

Figure 1 presents a simplified architecture of the Cloud XR system considered in the paper. The XR scene is rendered at the cloud server, encoded into a video stream with a specific visual quality, and transmitted to the XR-headset frame-by-frame. The headset presents the scene by playing the video to the user, captures her actions with the sensors, and generates

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

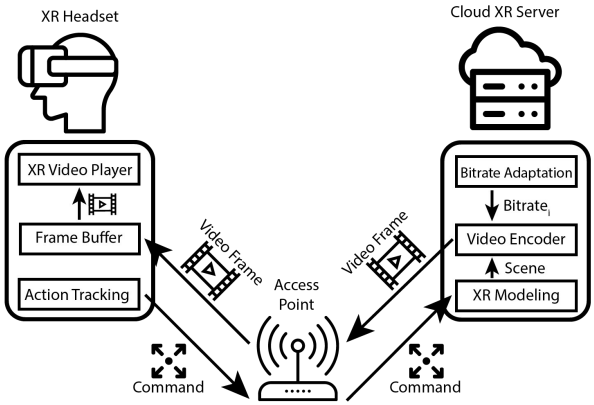

Figure 1. A sample Cloud XR architecture

the control commands for the server. The server receives the commands, updates the scene accordingly, and transmits the next frames to the headset. Such a workflow is typical for the existing cloud-based gaming and XR systems [10, 33, 34].

Because the scene changes according to the user actions, the sooner the commands are delivered, the less is the scene refresh delay perceived by the user. Therefore, in the network, the commands shall have higher priority than the video stream.

The video stream is the sequence of video frames generated by the server with the inter-frame interval T. The size of each frame is $S = b \cdot T$, where b is the bitrate of the video stream. In the paper, we assume that the higher is the bitrate, the higher is the video quality, which is the typical case for a well-engineered streaming system [35].

The XR scene presentation at the headset is organized as the following video playback. The client builds up the initial playback buffer of K_0 video frames, and only after that, it starts the playback. The larger is K_0, the more resilient is the playback to the variations in the frame delivery rate, but at the same time, the higher is the scene refresh delay perceived by the user. The client pulls one whole frame from the playback buffer every interval T. If there is no whole video frame in the buffer, the client does not play anything, and a video stall occurs. At this point, the virtual buffer of the server, i.e., the number of video frames not yet delivered to the client, contains K_0 frames. To reduce the load on the network and to keep the delay between frame generation and playback limited, the server discards the next generated frame. This way, the server and client buffer levels have a one-to-one correspondence. The pseudocode of the video playback algorithm is presented in Algorithm 1.

In the paper, we assume that the server knows the exact amount of data not yet delivered to the client. Since the server needs this information once in the inter-frame interval, it can be easily achieved in practice. For example, the client can send its current buffer level to the server with the control commands that shall be delivered with a small delay anyway. Alternatively, to obtain the most relevant information on the

Algorithm 1 XR video playback algorithm

▷ K is the XR session duration (in inter-frame intervals)
▷ $rxFrames$ is the list of frames in playback buffer
▷ $ReceiveFrame$ is the function to receive data from the network
$playbackStarted = False$
while $NOT\ \ playbackStarted$ **do**
    $frame = ReceiveFrame()$
    $rxFrames.push(frame)$
    if $length(rxFrames) >= K_0$ **then**
        $PlaybackStarted = True$
    end if
end while
$playbackFrame = 0$
for all $k \in 1, \bar{K}$ **do**
    $frame = ReceiveFrame()$
    if $IsFullFrame(frame)$ **then**
        $rxFrames.push(frame)$
    end if
    if $length(rxFrames) > 0$ **then**
        $playbackFrame = rxFrames.pull()$
    else
        $playbackFrame = 0$
    end if
    $Play(playbackFrame)$
end for

undelivered data, the server can use the dedicated control connections with the routers along the data transmission path [36].

In the network, in addition to the command traffic the other delay-sensitive traffic can have a higher priority than the video flows. The traffic can be generated by factory automation, telemedicine, autonomous and remote driving applications. Multiple sources with different patterns generate random high-priority traffic, and the user's actions causing the command traffic are unpredictable too. Therefore, we model the high-priority traffic as a Poisson flow of packets with the rate λ and a general packet size distribution.

In the paper, we assume that the high-priority traffic receives no interference from the XR video flow. Hence, we can estimate its QoS independently using well-known analytical results for M/G/1-type queueing systems (e.g., see [37, 38]).

The channel resources consumed by the high-priority traffic during an inter-frame interval change with time. Therefore, to reduce the probability of stalling and maximize the video quality, the bitrate of the video stream shall be adjusted adaptively according to the amount of channel resources remaining after servicing the high-priority traffic. Further, in the paper, we consider the following bitrate adaptation scheme, which is similar to the bitrate adaptation performed by the cloud gaming service Google Stadia [33]. Once in an inter-frame interval the server analyzes the current virtual buffer state and chooses the bitrate of the next generated frame from the discrete set $\mathcal{B} = \{b_0, b_1, ..., b_{N_b}\}$, where

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

$0 = b_0 < b_{min} \leq b_1 \leq ... \leq b_{N_b} = b_{max}$. Therefore, the bitrate adaptation function can be an arbitrary function of the buffer state. The pseudocode of the considered scheme is presented in Algorithm 2.

Algorithm 2 XR bitrate adaptation scheme

```
▷ K is the XR session duration (in inter-frame intervals)
▷ framesInTx is the list of non-delivered frames
▷ GetBitrate is the bitrate adaptation function
▷ GenerateFrame is the frame rendering function
for all k ∈ {1, ..., K} do
    for all frame ∈ framesInTx do
        if IsFrameDelivered(frame) then
            framesInTx.remove(frame)
        end if
    end for
    bitrate = GetBitrate(framesInTx)
    frame = GenerateFrame(bitrate)
    SendFrame(frame)
    framesInTx.push(frame)
end for
```

To sum up, the considered system can be represented as a queuing system with two queues:

1) The M/G/1 queue with high-priority traffic (absolute-priority queue),
2) The D/G/1 queue with video frames (low-priority queue).

We assume that the service provided to the XR video flow in one inter-frame interval of duration T is independent of the service in the previous intervals. Therefore, the evolution of the virtual server buffer state can be modeled with the discrete-time Markov chain with time unit T. The transition probabilities depend on the consumption of channel resources by the high-priority traffic during a single time unit and on the bitrate adaptation function.

B. PROBLEM STATEMENT

In the paper, we address the problem of QoE modeling and optimization for adaptive Cloud XR application traffic in the presence of interfering high-priority traffic in the network. In the considered scenario, the QoE of the Cloud XR applications can be reduced to the QoE of the real-time adaptive video streaming, which is a complex subjective metric. Its models usually take into account the factors from various parts of the video streaming system. They include the video encoder, its parameters (e.g., frame rate, frame structure, bitrate, etc.), the parameters of the network connection (e.g., capacity, delay, packet loss ratio), and the properties of the playback device (e.g., screen size and resolution and frame rate) [32].

In the paper, we focus on such QoS-derived QoE metrics as the average video bitrate and the stalling probability [39]. We choose these metrics because they provide a good trade-off between the QoE modeling accuracy and the estimation complexity. We develop the model of the considered system and use it to estimate the QoE of the XR video flows for various buffer-state-based bitrate adaptation functions. We select a family of bitrate adaptation functions and use the model to find the optimal function. Namely, we find the function that maximizes the average video bitrate for a particular high-priority traffic rate and subject to a constrained stalling probability. We show that while the state of the network is not stationary and the rate of the high-priority traffic can change with time, we can obtain the optimal adaptation functions for a range of rates. Consequently, the server applies the bitrate adaptation functions according to the perceived high-priority traffic rate. We assume that this rate can be measured by the network and communicated to the server via a cross-layer protocol (e.g., [36, 40]).

III. LITERATURE REVIEW

In the paper, we model the QoE of the XR traffic as a QoE for the real-time (i.e., delay-sensitive) adaptive video streaming in the presence of the interfering high-priority traffic. We aim to develop an adaptation algorithm for the Cloud XR video streaming that will provide an optimal QoE. Therefore, in Section III-A, we review the prior arts in the video quality adaptation and show why we needed to develop our model. Because the model is based on the queueing theory, in Section III-B, we survey the relevant results from this area. We demonstrate that, despite it is a well-researched area, to the best of our knowledge, some of the M/G/1 queue characteristics we obtain in this paper are novel.

A. VIDEO QUALITY ADAPTATION

Video quality adaptation is a well-researched area, and many papers develop different adaptation schemes in an attempt to address certain problems and improve users' QoE. However, most of the papers study the video quality adaptation problem in the framework of HTTP Adaptive Streaming (HAS) [39]. For example, a well-known algorithm is proposed in [41]. This algorithm is implemented in Dash.js, a reference open-source video player for the MPEG-DASH technology [42]. The algorithm optimizes a utility function of stall frequency and the average bitrate of the video stream. The authors show that this algorithm is optimal for infinite video streams. The paper [43] further improves the performance of the algorithm in the case of live HAS. Another well-known bitrate adaptation algorithm is proposed in [44]. This algorithm serves as a basis for the bitrate adaptation algorithm implemented by Netflix. Similar to the previous one, it uses the video buffer occupation as the main factor for the bitrate adaptation. A control-theoretic approach to the bitrate adaptation algorithm design is employed in [45]. The authors formulate the bitrate adaptation as a control problem and develop an algorithm aimed at reducing the video buffering while maximizing the video bitrate. With simulations, the authors show that the proposed algorithm outperforms the state-of-the-art ones.

These algorithms were mostly designed for video-on-demand streaming, where a client can pre-buffer a large amount of video to efficiently smoothen the network capacity

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

fluctuations. However, the bitrate adaptation for real-time video streaming is an even more challenging task because the algorithms have a much lower policy space. So, many recent papers build artificial neural networks so that they could find the optimal bitrate adaptation scheme. In [46], the authors develop an algorithm for joint bitrate and buffer control for low-latency video streaming. In the proposed scheme, the algorithm dynamically adjusts both the bitrate of the downloaded video and the playback rate to reduce the probability of video stalling. With simulations, the authors show that the proposed algorithm provides higher QoE than state-of-the-art in low-latency streaming scenarios. Another neural-network-based algorithm is proposed in [47]. The algorithm is designed for video streaming for the remote control of unmanned aerial vehicles. The algorithm takes into account the fluctuations of the air-to-ground channel capacity and predicts the channel capacity to stream video appropriately. To further reduce the presentation latencies, the paper [48] proposes splitting the frames into subframes and encoding and sending them independently. This way, the client can receive and start decoding the parts of frames earlier. However, state-of-the-art video codecs do not support such a technique, so it is difficult to implement in practice.

Unfortunately, most of the papers analyze the performance of the algorithms with simulations or by considering the asymptotical cases. So, to develop an understanding of the QoE limits of the adaptive XR video streaming in the presence of high-priority interfering traffic, further we address the problem from the queueing theory perspective. Specifically, as stated in Section II, we need to find the probability distribution of the amount of resource consumption by the M/G/1 queue in a fixed time interval.

B. PRIORITY QUEUES AND M/G/1-TYPE SYSTEMS

The queueing systems with multiple priority queues have been well-studied in the literature for a rather long time. In [49], the authors investigate various ways to organize queues with priorities and model the transients for these queues. They consider a combination of two or more queues of type M/G/1. The authors calculate the distribution of the duration of the continuous busy period of the M/G/1 queue and the probability of the queue being busy at an arbitrary time moment. Also, as a straightforward derivation, they obtain the average resource consumption in the M/G/1 queue at the finite time interval. However, the occupation time distribution at the finite interval is not calculated, so the results of [49] are not applicable to solve the problem considered in our paper.

The paper [50] describes a mathematical model for joint service of web and MPEG-DASH video traffic. The system in consideration is a system with two queues where one is a high-priority M/D/1 queue of web-pages. The distribution of service-free time over a fixed length interval is calculated. However, in our case, the packet service time is not constant. Therefore, the distribution obtained in [50] cannot be applied to solve the problem.

In our model, the service provided to the low-priority D/G/1 XR video queue depends on the occupation time probability distribution of the high-priority M/G/1 queue during a short interval. Note that for the infinite observation interval, the server occupation time in an M/G/1 system has been studied in detail in the literature. Existing works provide well-known methods to obtain such important long-term characteristics as the average waiting time [37], average queue length [51], average busy period duration [52]. However, these characteristics were not well-studied for the interval of finite duration.

Researchers have also considered various characteristics of the system at finite time intervals. In the classic work [53], the M/G/1 system was considered with an additional condition for customers entering the queue. If a customer arrives in the queue when the server is busy, it leaves the queue with a certain probability. In this case, the transient processes of the system were investigated, and service characteristics such as the distribution of virtual waiting time and the average duration of the busy period were obtained. The virtual waiting time is the time required to release the system from servicing requests that have arrived before a particular moment. Transient processes for the classic M/G/1 queue were investigated in the paper [54]. The authors also focused on the virtual waiting times. In this work, the time-dependent server-occupation probability and a virtual waiting period were obtained. However, the results for the virtual waiting period do not apply to the problem addressed in our paper.

The distribution of the busy period of the M/G/1 queue was obtained in paper [55]. The authors consider not only the states of the system when it is busy or free from servicing requests, but also the general case: when there are no more (or vice versa, more) than a certain number of requests in the system. For the time interval tending to infinity, asymptotic distributions of times spent by the system in these states have been obtained. Moreover, at the beginning of the time interval under consideration, only one boundary case is taken into account: the absence of requests. The resulting asymptotic distributions of the modified system [55] cannot be applied to our study because, in the considered system, the time intervals are short, and the use of asymptotic is not possible.

Although the probabilistic properties of the M/G/1 queues are well-studied, to the best of our knowledge, no method exists to calculate the required finite interval occupation time distribution for the M/G/1 queue. So, we develop such a method in Section IV-B.

IV. ANALYTICAL MODEL

This section develops an analytical model of the heterogeneous traffic service: real-time adaptive video traffic, namely, XR scene streaming, and control traffic. In particular, in Section IV-A, we discuss the design of the Markov chain modeling the video buffer state evolution. In Section IV-B, we estimate the probability distribution function of the resource consumption by the high-priority traffic. Finally, in Section IV-C, we describe the proposed bitrate adaptation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

algorithm optimization framework. Table 1 summarizes the main notations used in the model.

A. VIDEO BUFFER STATE EVOLUTION

We define $\mathbb{S}$ as a set of all possible states of the server virtual buffer (hereinafter, the buffer) and describe each state $S \in \mathbb{S}$ with a (K_0+1)-dimensional vector $S = (j, i_1, ..., i_{K_0})$. Here, the first index, j, indicates $q_j \in \mathcal{Q} = \{q_0, q_1, ..., q_{N_q}\}$, where q_j is the discretized fraction of the last frame partially received by the client and $0 = q_0 \leq q_1 \leq ... \leq q_{N_q} = 1$. The indices i_r are the indices of bitrates $b_{i_r} \in \mathbb{B}$ of the video frames in the buffer.

Assuming that the service provided to the video flow in one inter-frame interval of duration T is independent of the service in the previous intervals, we can describe the buffer state evolution with a discrete-time Markov chain with the time unit T. The states of the chain are the states of the buffer at the time moments of the next video frame bitrate choice. The chain is aperiodic and irreducible, which leads to its ergodicity and the existence of stationary distribution $\pi(S)$. Next, we describe all possible transitions between the states of the chain and estimate the transition probabilities.

If $S = (j, i_1, ..., i_n, 0, ..., 0)$, then the new video frame is generated with the bitrate $b_{i_{n+1}} = B(S)$, where $B(X)$ denotes the bitrate adaptation function. For $n < K_0$, the following transitions from S are possible:

1) A part of the first video frame is transmitted and a new video frame is generated:
   $S \to (l, i_1, ..., i_n, i_{n+1}, 0, ..., 0)$, for $1 \leq l \leq j$.
2) The first few video frames are transmitted and a new video frame is generated:
   $S \to (l, i_m, ..., i_n, i_{n+1}, 0, ..., 0)$, for $1 < m \leq n, 1 \leq l \leq N_q$.
3) All the video frames are transmitted and a new video frame is generated:
   $S \to (N_q, i_{n+1}, 0, ..., 0)$.

If $S = (j, i_1, ..., i_{K_0})$ and $i_{K_0} > 0$, then the new video frame is not generated and $B(S) = 0$. In this case, the following transitions are possible:

1) A part of the first video frame is transmitted and a new video frame is not generated:
   $S \to (l, i_1, ..., i_{K_0})$, for $1 \leq l \leq j$.
2) The first few video frames are transmitted and a new video frame is not generated:
   $S \to (l, i_m, ..., i_{K_0}, 0, ..., 0)$, for $1 < m \leq K_0, 1 \leq l \leq N_q$.
3) All the video frames are transmitted:
   $S \to (0, ..., 0)$.

If $S = (0, ..., 0)$, the buffer is empty, $b_{i_1} = B(0, \cdots, 0)$, and only one transition is possible:

1) A new video frame is generated with $b_{i_1} = B(0, \cdots, 0)$:
   $S \to (N_q, i_1, 0, ..., 0)$.

Let $h(x)$ be the probability density function (p.d.f.) of the event that an arbitrary time interval T has enough free time for transmission of exactly x bytes of the video flow. We obtain $h(x)$ from $e(t)$, which is the p.d.f. of the event that in an M/G/1 queue during time T the interval of duration t is occupied:

$$h(x) = e\left(T - \frac{x}{R}\right), \tag{1}$$

where R is the rate of data transmission. We derive $e(t)$ in Section IV-B.

We fix a certain state S of the considered Markov chain and estimate the buffer level $V(S)$ in the state S as:

$$V(S) = q_j \cdot b_{i_1} \cdot T + T \sum_{w=2}^{K_0} b_{i_w}. \tag{2}$$

Let $\vec{M}$ be the set of all states M_i of the chain for which the values of transition probability from the state S are positive: $\vec{M} = \{M_1, ..., M_{N(S)}\}$, where $N(S)$ is the power of set $\vec{M}$. The amount of transmitted data $V_S^{M_i}$ for the chain to transit from the state S to the state M_i is:

$$V_S^{M_i} = V(S) - (V(M_i) - B(S) \cdot T).$$

This amount of transmitted data depends on the time free from serving a high-priority queue during a period T.

The discretization of the share $q \in \mathcal{Q}$ allows us to estimate the probability of transition from the state S to the state M_i as the probability of transmission of an arbitrary number of bytes from the interval including $V_S^{M_i}$. Such intervals should not intersect and should cover the whole set $[0, V(S)]$. Therefore, the boundaries of the intervals $[m(i-1), m(i)]$ are $m(i) = \frac{V_S^{M_i} + V_S^{M_{i+1}}}{2}$. Thus, we obtain the transition probability from the state S to the state M_i:

$$p(S, M_i) = \begin{cases} \int_0^{m(1)} h(x)\, dx, & i = 1; \\ \int_{m(i)}^{m(i+1)} h(x)\, dx, & 1 < i < N(S); \\ 1 - \sum_{l=1}^{N(S)-1} p(S, M_l), & i = N(S). \end{cases} \tag{3}$$

Finally, we obtain the stationary probability distribution π by solving the following system of linear equations:

$$\sum_{S' \in \mathbb{S}} \pi(S') p(S', S) = \pi(S), \forall S \in \mathbb{S},$$

$$\sum_{S' \in \mathbb{S}} \pi(S') = 1.$$

A stall during playback occurs if the buffer contains K_0 video frames. Therefore, to estimate the stall probability P_{stall}, we need to assess the probability that the chain is in the states with $i_{K_0} \neq 0$:

$$P_{stall} = \sum_{i_{K_0} \neq 0} \pi(j, i_1, \cdots, i_{K_0}). \tag{4}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

Table 1. Key notation.

λ	the high-priority traffic intensity
$F(t)$	the high-priority service time c.d.f.
T	the video inter-frame interval duration
$\mathbb{S}$	the set of all possible buffer states
$B(S)$	bitrate adaptation function, where S is the state of the buffer
$h(x)$	p.d.f. of the event that an interval T has enough free time to transmit x bytes of the video
$e(t)$	p.d.f. of the event that in M/G/1 queue, the sum of busy periods equals t during an interval T
$\pi(S)$	the stationary distribution of a discrete-time Markov chain of buffer state evolution
$p(S, M_i)$	the transition probability from the state S to the state M_i
P_{stall}	the stall probability
B_{av}	the average bitrate
$p_{T'}(x)$	p.d.f. of occupation time for the main stage of duration T'
$l(t)$	p.d.f. of the starting stage duration
P_{border}	the probability of the presence of the starting stage
$P_{start>T}$	the probability that the starting stage lasts for the entire interval T

The average bitrate B_{av} of the video is estimated as:

$$B_{av} = \sum_{S} \pi(S) B(S). \tag{5}$$

B. OCCUPATION TIME DISTRIBUTION OF THE M/G/1 SYSTEM IN A FIXED TIME INTERVAL

System occupancy can be described with an ON/OFF process generated by the alternating busy and empty periods of the M/G/1 queue. Without loss of generality, we may consider that the interval T starts at time 0. We consider the following stages of the system evolution during the interval:

- The starting stage. The starting stage is a part of the busy period of the system that starts before time 0 and spans over the beginning of the time interval.
- The main stage. The main stage is an ON/OFF process with the boundary condition: the stage starts with an OFF-period.

1) Distribution of ON- and OFF-period duration

Let $f_{ON}(t)$ and $f_{OFF}(t)$ be the p.d.f. of ON- and OFF-periods, respectively. The duration of an OFF-period is distributed exponentially with the mean $1/\lambda$. For M/G/1 system with the service time cumulative distribution function (c.d.f.) $F(t)$, the c.d.f. $F_{ON}(t)$ of the busy (ON) period duration is determined according to [56] as:

$$F^*_{ON}(s) = F^*[s + \lambda - \lambda F^*_{ON}(s)], \tag{6}$$

where $F^*(s)$ is the Laplace-Stieltjes transform of the service time c.d.f. $F(t)$.

Let $t^{(k)}_{OFF}$ and $t^{(k)}_{ON}$ be the sum of k OFF-periods and ON-periods, respectively. The p.d.f. of such sums can be calculated with convolution and we denote them as $p^{(k)}_{OFF}(t)$ and $p^{(k)}_{ON}(t)$, where $p^{(0)}_{ON}(t) = p^{(0)}_{OFF}(t) = \delta(t)$ and $\delta(t)$ is the Dirac delta function.

2) Occupation time at the main stage

The main stage starts with an OFF-period and has a duration $T' <= T$. We calculate the distribution of the occupation time of such a stage.

The probabilities that $k-1$ OFF(ON)-periods have the total duration less than T' and k OFF(ON)-periods have the total duration longer than T' are evaluated as:

$$\mathbb{P}\left(\left(t^{(k-1)}_{OFF} < T'\right) \,\&\, \left(t^{(k)}_{OFF} > T'\right)\right) = \\ = \mathbb{P}\left(t^{(k-1)}_{OFF} < T'\right) - \mathbb{P}\left(t^{(k)}_{OFF} < T'\right), \\ \mathbb{P}\left(\left(t^{(k-1)}_{ON} < T'\right) \,\&\, \left(t^{(k)}_{ON} > T'\right)\right) = \\ = \mathbb{P}\left(t^{(k-1)}_{ON} < T'\right) - \mathbb{P}(t^{(k)}_{ON} < T').$$

For $k = 1$, the expressions take the following forms: $\mathbb{P}\left(t^{(k)}_{OFF} > T'\right)$ and $\mathbb{P}\left(t^{(k)}_{ON} > T'\right)$.

Then $\forall k \in \mathbb{N}$, we can calculate p^{idle}_k, the probability that k ON-periods have a total duration of x and k OFF-periods have a total duration of less than $T' - x$, and $k+1$ OFF-periods have a total duration of more than $T' - x$. In other words, exactly time x of the main stage is occupied and the system is idle at the end of the main stage:

$$p^{idle}_k(x) = p^{(k)}_{ON}(x) \cdot \\ \cdot \mathbb{P}\left[\left(t^{(k)}_{OFF} < T' - x\right) \,\&\, \left(t^{(k+1)}_{OFF} > T' - x\right)\right].$$

Similarly, we calculate p^{busy}_k, the probability that k OFF-periods have a total duration of $T' - x$ and $(k-1)$ ON-periods have a total duration of less than x, and k ON-periods have a total duration of more than x. In other words, exactly time x is occupied and the system is busy at the end of the main stage:

$$p^{busy}_k(x) = p^{(k)}_{OFF}(T' - x) \cdot \\ \cdot \mathbb{P}\left[\left(t^{(k-1)}_{ON} < x\right) \,\&\, \left(t^{(k)}_{ON} > x\right)\right].$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

Finally, we obtain the occupation time p.d.f. for the main stage of duration T'. We denote it as $p_{T'}(x)$:

$$p_{T'}(x) = \sum_{k=1}^{\infty} \left(p_k^{idle}(x) + p_k^{busy}(x) \right).$$

3) Distribution of the starting stage duration

The starting stage duration is the remaining duration of an ON-period of an ON/OFF process from an equiprobably chosen starting point $t_0 = 0$. In [57], we prove that this remaining duration has the following p.d.f.:

$$l(t) = \frac{1 - F_{ON}(t)}{< T^{ON} >},$$

where $F_{ON}(t)$ is the c.d.f. of the ON-period duration and $< T^{ON} >$ is its expected value.

The starting stage can be absent, so let us denote the probability of its presence as P_{border}. For the considered ON/OFF process, this probability equals the probability that an arbitrary random point on the time axis belongs to the ON-period. So, it can be estimated as the share of time when the system is occupied:

$$P_{border} = \frac{< T^{ON} >}{< T^{ON} > + < T^{OFF} >}, \tag{7}$$

where $< T^{OFF} >= \frac{1}{\lambda}$ is the average duration of the OFF-period.

Finally, we evaluate $P_{start>T}$, the probability that a starting stage lasts for the entire time interval T:

$$P_{start>T} = 1 - \int_0^T l(t)\, dt. \tag{8}$$

4) System occupation time distribution

We estimate $e(t)$, the probability that the system is occupied for t, $0 \le t \le T$, by considering the contributions of the following system evolution cases:

1) Only starting stage is present:

$$P_{start>T} \cdot P_{border} \cdot \delta(T - t).$$

2) Both starting and main stages are present:

$$P_{start \le T} \cdot P_{border} \cdot \tilde{e}_1(t), \text{ where}$$
$$\tilde{e}_1(t) = \int_0^t l(\tau) p_{T-\tau}(t - \tau) d\tau.$$

3) Only the main stage is present:

$$(1 - P_{border}) \cdot p_T(t).$$

By summing the above, we obtain:

$$\begin{aligned} e(t) = P_{start>T} \cdot P_{border} \delta(T - t) + \\ + P_{start \le T} \cdot P_{border} \tilde{e}_1(t) + \\ + (1 - P_{border}) p_T(t). \end{aligned}$$

Using $e(t)$, the p.d.f. $h(x)$ of the event that inside an interval T there is sufficient free time for video transmission of x bytes can be found with (1). This function is used to obtain the transition probabilities $p(S, M_i)$ with (3) and, subsequently, the stationary distribution of Markov chain $\pi(S)$. Finally, it allows us to estimate the stall probability (4) and the average bitrate (5).

C. OPTIMIZATION OF THE BITRATE ADAPTATION FUNCTION

We use the developed model to optimize the video bitrate adaptation function. The function can take as an input the buffer level or more detailed information on the buffer state: the number of video frames in the buffer, their average bitrate, etc.

Let us consider a function space $\mathcal{F}$ of bitrate adaptation functions. The average bitrate and the stalling probability can be defined as functions of the scenario parameters and the bitrate adaptation function: $B_{av} = B_{av}(B, G, \lambda, ...)$ and $P_{stall} = P_{stall}(B, G, \lambda, ...)$, where $B \in \mathcal{F}$. Thus, we can introduce the following optimization problem. For particular scenario parameters, we need to find such bitrate adaptation function that maximizes the average bitrate of the video stream and guarantees the limited video stalling probability θ:

$$B^{(opt)} = \arg\max_{B \in \mathcal{F}} \{ B_{av}(B) : P_{stall}(B) \le \theta \}. \tag{9}$$

In the paper, we consider a space of piecewise constant bitrate adaptation functions. We define the current buffer level U as the total amount of bytes not yet delivered to the client. Then, we can define the bitrate adaptation function as $B(U) = B\left(\frac{U}{U_{max}}\right)$, where $U_{max} = b_{max} T(K_0 - 1)$. For a fixed number of bitrate levels N, the piecewise constant function $B(U)$ is defined by the set of reference values of bitrates $\vec{B} = [B_1, ..., B_N]$ Mbps and the relative buffer levels $\vec{U} = [U_1, ..., U_{N-1}]$, so that:

$$B(U) = \begin{cases} B_1, & \frac{U}{U_{max}} \le U_1; \\ B_i, & U_{i-1} < \frac{U}{U_{max}} \le U_i; \\ B_N, & \frac{U}{U_{max}} > U_{N-1}. \end{cases}$$

The pseudocode of the considered bitrate adaptation function is presented in Algorithm 3.

The output of the optimization is such a piecewise constant bitrate adaptation function defined by the set $(\vec{B}_{opt}, \vec{U}_{opt})$ for which

$$(\vec{B}_{opt}, \vec{U}_{opt}) = \arg\max_{\vec{B}', \vec{U}'} \{ B_{av}(\vec{B}', \vec{U}') : P_{stall}(\vec{B}', \vec{U}') \le \theta \}. \tag{10}$$

We carry out the optimization numerically and discuss its results in the next section.

V. NUMERICAL RESULTS

In this section, we use the well-known network simulation platform ns-3 [58] to validate the model and demonstrate the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

Algorithm 3 Piecewise-constant bitrate adaptation function

Input: $\vec{B} = [B_1, ..., B_N], \vec{U} = [U_1, ..., U_{N-1}], \mathbb{B}$

function $GetBitrate(framesInTx)$
  $U = 0$
  $b_{max} = \max\{b_i \in \mathbb{B}\}$
  $U_{max} = (K_0 - 1) \cdot T \cdot b_{max}$
  for all $frame \in framesInTx$ **do**
    $U += frame.size()$
  end for
  $U_{relative} = \frac{U}{U_{max}}$
  $i^* = \arg\max_i \{U_{relative} \leq U_i\}$
  return B_{i^*}
end function

results of the developed method of bitrate adaptation function optimization. In Section V-A, we describe the considered scenario. Then, in Section V-B, we use the model to estimate the network capacity for the XR video. In Section V-C, we present the results of the bitrate adaptation function optimization. Finally, in Section V-D, we present and discuss the results of the model validation and compare QoE provided by the optimized bitrate adaptation function with one of the state-of-the-art.

A. SCENARIO

We consider a basic Cloud XR scenario with an XR-user playing an XR-game at home. To provide freedom of movement and, thus, an immersive experience to the user, the headset uses Wi-Fi and connects to a remote Cloud XR server via a Wi-Fi access point. The access point has a wired connection with the Cloud XR server. To minimize the feedback delay, the XR-headset sends the commands to the server using channel access parameters corresponding to high priority access category AC_VO. To avoid interference with the commands, the AP sends video frames to the headset using the low priority access category AC_BK.

We model the high-priority command traffic as a stream of packets or bursts of packets with an exponentially distributed size in bytes with mean μ. We choose the average command size $\mu = 90$ kB, so that, in the considered scenario, the average transmission time of one command is 1 ms. The rate of commands varies in the range $[0.05, 0.3]$ ms^{-1}. The server generates an XR video stream with a bitrate in the range of 8 to 72 Mbps. This corresponds to the image visual quality ranging from a typical home cinema FullHD to a UHD panoramic video with a high frame rate. The period of video frame generation is $T = 15$ ms. Other scenario parameters are given in Table 2.

Using both the developed model and simulations, we estimate the QoE of the XR user with the following metrics.

1) *Average bitrate:* the average bitrate of the XR video stream during the experiment, i.e., the average size of the video frame divided by the inter-frame interval duration.

Table 2. Scenario parameters.

Parameter	Value
Wi-Fi standard	802.11ac
Carrier frequency	5 GHz
Channel width	80 MHz
MCS	VhtMCS3
Pathloss model	IEEE TGax Simulation Scenarios [59]
AP/STA Tx Power	30 dBm
AP/STA height	1.5 m
Packet aggregation	A-MPDU
$AIFSN$(AC_BK)	7
CW_{min}(AC_BK)	15
CW_{max}(AC_BK)	1023
$AIFSN$(AC_VO)	2
CW_{min}(AC_VO)	3
CW_{max}(AC_VO)	7
MTU	1460 Bytes
Experiment duration	1000 s
Number of experiment runs	40
Set of bitrates $\mathbb{B}$	$\{8, 16, 24, 32, 40, 48, 56, 64, 72\}$ Mbps
Average command size, μ	90 kB
Jitter-buffer depth, K_0	3 frames
Stall probability limit θ	0.01
Frame generation period T	15 ms
First frame portions $\mathcal{Q}$	$\{0, 0.1, 0.2, ..., 0.9, 1\}$

2) *Stall probability:* the portion of inter-frame intervals when the client does not have a whole video frame in the video buffer.

B. ESTIMATION OF THE NETWORK CAPACITY FOR CLOUD XR

In this section, we use the model to estimate the network capacity for non-adaptive XR video stream in the considered scenario. Specifically, for a given high-priority traffic intensity, we find the maximal XR video bitrate, for which the stalling probability is less than $\theta = 0.01$. To illustrate the advantages of the developed model, we compare its results with the following network capacity estimation representing the average channel capacity available to the video stream:

$$C^{(0)} = C\left(1 - \frac{\lambda}{\mu}\right), \tag{11}$$

where C is the average capacity of the channel between the headset and the access point ($C = 75$ Mbps in the considered scenario).

Figure 2 presents the network capacity for the XR video stream (i.e., the bitrate of the video stream) for each command rate. The results show that the actual capacity of the network for the XR video is up to 50% lower than we can obtain with (11). This happens because eq. (11) considers

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

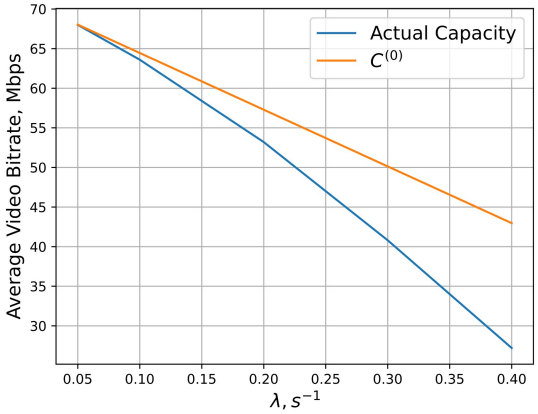

Figure 2. Network capacity estimation.

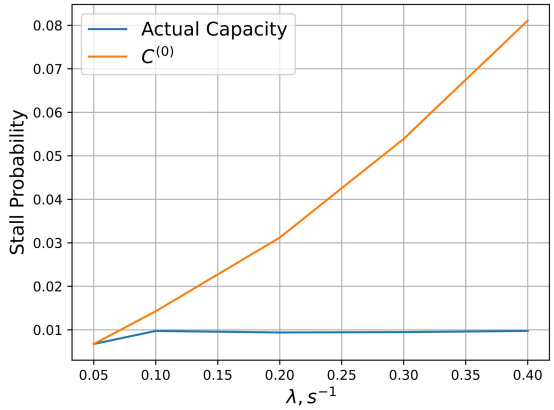

Figure 3. Network capacity estimation. Stall Probability.

the average values instead of the probability distributions. The difference between the actual capacity and C_0 increases with λ because the variance of the resource consumption by the commands grows with the command rate. Also, this significantly increases the probability that a video frame is not delivered during K_0 inter-frame intervals. Consequently, Figure 3 shows that XR video streams with bitrate $C^{(0)}$ have up to eight times higher stalling probability than the considered QoE requirement θ.

C. BITRATE ADAPTATION FUNCTION OPTIMIZATION

In this section, we use the model to find the optimal for the considered scenario bitrate adaptation functions from the space of piecewise constant functions with the number of bitrate levels $N = 4$. The admissible bitrates and relative buffer levels are chosen from the sets $\vec{B}_{pool} = [8, 16, 24, ..., 72]$ Mbps and $\vec{U}_{pool} = [0, 0.1, 0.2, ..., 1]$ respectively. We set the constraint on the stalling probability $\theta = 0.01$, i.e., the optimal bitrate adaptation algorithm shall maximize the average video bitrate while losing less than 1% of frames.

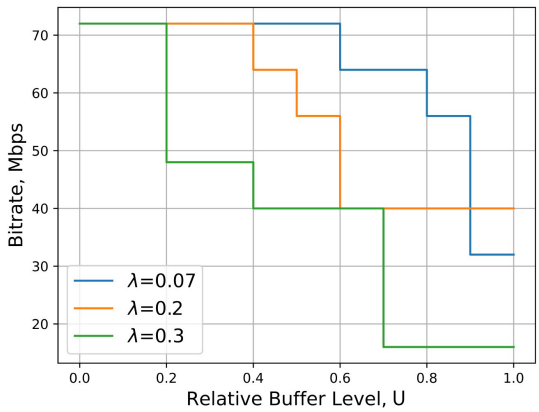

Figure 4. Optimal piecewise constant bitrate adaptation functions.

Taking into account that the bitrate adaptation functions of the relative buffer level should be non-increasing, we search through the function space. For each of the bitrate adaptation functions, we estimate the average video bitrate and stall probability and determine the optimal combination $(\vec{B}^{(opt)}, \vec{U}^{(opt)})$ for each command rate according to (10). The considered optimization scheme requires calculating the model with $N_{eval} = C^{N}_{N-1+|\vec{B}_{pool}|} \cdot C^{N-1}_{N-2+|\vec{U}_{pool}|}$ sets of parameters, where $|\cdot|$ represents the cardinality of the set. In turn, a single model calculation requires solving the linear system of $N_{eq} = (\sum_{i=1}^{K_0} N^i) \cdot |\mathcal{Q}| + 1$ equations. Although the resulting optimization problem appears to be rather complex, in practice, we do not have to solve it online, because we can pre-calculate the optimal functions for a range of scenarios.

Figure 4 shows the optimal piecewise constant bitrate adaptation functions $B^{(opt)}_{\lambda}(U)$ for the command rates $\lambda = [0.07, 0.2, 0.3]$ ms^{-1}. From the figure, we can see that for all considered loads, the optimal adaptation functions choose the highest available bitrate for some range of buffer levels. This means that, even for $\lambda = 0.3$, there is only a small probability that the frame with the maximal bitrate is delivered longer than $(K_0 - 1)$ inter-frame intervals. Another effect is caused by a limited number of function steps. On the one hand, the average bitrate for the function $B^{(opt)}_{\lambda=0.07}$ is higher than for $B^{(opt)}_{\lambda=0.2}$. On the other, if the buffer is almost full, the former chooses a lower bitrate than the latter. The reason for this is the limited number of function steps. To meet the stalling probability requirement, $B^{(opt)}_{\lambda=0.2}$ chooses rather a low bitrate in advance (when $U = 0.6$) and does not decrease it further. However, if the number of function steps was larger, at $U = 0.6$, the optimal function would choose a higher bitrate but would decrease it further at a larger U.

To find an optimal bitrate adaptation function for the range of command rates, we aggregate the optimal bitrate adaptation functions for each λ and obtain $B^{(opt)}(U) = B^{(opt)}_{\lambda}(U)$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

D. COMPARISON OF VARIOUS BITRATE ADAPTATION ALGORITHMS

We consider the following adaptation algorithms:

1) A_OPT: the adaptation algorithm described in Section IV-C with the bitrate adaptation function $B^{(opt)}(U)$.
2) A3: the adaptation algorithm described in Section IV-C with the bitrate adaptation function $B^{(opt)}_{\lambda=0.3}(U)$. We consider this bitrate adaptation function because it shall provide satisfactory stall probability in the considered range of command rates, but it requires finding a much smaller number of optimal parameters in comparison to the A_OPT algorithm.
3) BOLA: the algorithm developed in paper [43] and adapted for the case of real-time adaptive streaming. The parameters of the algorithm are set according to its reference implementation in [42]. Unlike other considered algorithms, we do not constrain the set of the bitrate levels for BOLA, so it can choose from the whole set $\vec{B}_{pool}$.

We compare the analytical and simulation results for algorithms A3 and A_OPT, but BOLA algorithm in live scenarios requires measuring the network throughput, so we estimate its performance only with simulations. Figures 5 and 6 present the target XR QoE metrics: the average bitrate and stall probability.

Let us start with the accuracy of the developed model. The figures show that at low rates of high-priority traffic (up to $\lambda = 0.2$), the developed model accurately describes QoE for the XR video stream. However, as the rate increases, the model starts underestimating the probability of stalling and accordingly overestimating the average bitrate of the video stream. This happens because the Markov property of the system disappears, i.e., the independence of the evolution of the video buffer from the consumption of resources by high-priority traffic in the previous periods of the video frame generation is lost.

Let us consider the probability $P^{frame}(T)$ that a command is delivered longer than the video inter-frame interval T, and the command is not displayed in the next generated frame. For the considered M/M/1 high-priority traffic model, this probability is calculated in [60]. It appears that at rates $\lambda > 0.23$, for the considered system $P^{frame}(T) \geq 10^{-5}$, which is the typical Packet Loss Ratio (PLR) requirement for the URLLC traffic [61]. So, the system shall not be used in such a regime.

Now let us analyze the QoE provided by different bitrate adaptation algorithms. We can see that the algorithm A_OPT allows significantly increasing the video bitrate at low rates of the high-priority traffic, and at the same time, fulfills the required stalling probability constraint. The provided results demonstrate the importance of optimizing the selection of bitrate for each high-priority traffic rate. While A3 fulfills the constraints on the probability of stalling at all considered rates, at low rates, it provides a 30% lower average bitrate

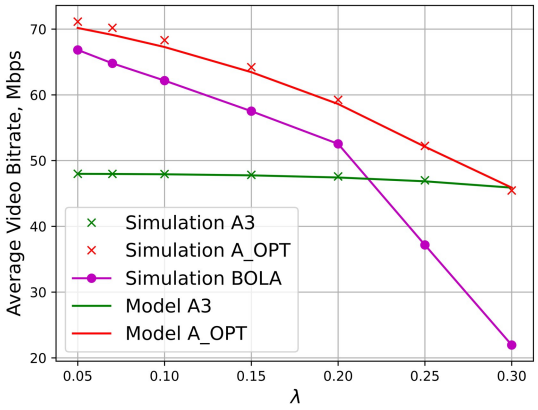

Figure 5. Comparison of the bitrate adaptation algorithms A3, A_OPT, BOLA. Average Bitrate.

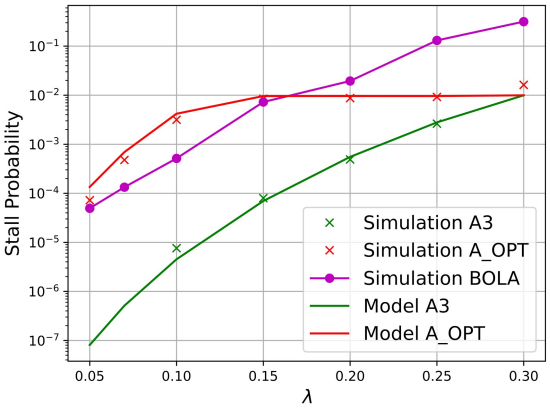

Figure 6. Comparison of the bitrate adaptation algorithms A3, A_OPT, BOLA. Stall Probability.

than the optimal algorithm. As for the BOLA, at low rates of high-priority traffic, the algorithm acts too conservatively and provides a lower average bitrate than the optimal algorithm. At the same time, at high rates, it cannot adapt to the fast changes of the network state and provides too high stall probability and up to 2 times lower average bitrate. So, we can conclude that BOLA cannot provide satisfactory QoE for Cloud XR streaming.

VI. CONCLUSION

In Cloud XR technology, rendering of the virtual scenes is performed at the remote server instead of the headsets. The server encodes the scenes into a real-time video stream and sends it to the headset. To provide higher QoE to the end-users, Cloud XR applications need to adapt to the changes in the network conditions and dynamically adjust the bitrate of the generated video stream. However, tight delay and high-reliability requirements significantly squeeze the room for bitrate adaptation optimization.

In the paper, we first studied analytically the problem of adaptive Cloud XR streaming in the presence of other types of high-priority real-time traffic, including the control traffic generated by the Cloud XR application itself. We designed a novel mathematical model of a real-time adaptive Cloud XR application. The model enabled us to estimate such QoE metrics for Cloud XR video stream as average bitrate and stall probability for a wide class of high-priority traffics: Poisson flow of packets with a general size distribution. With the model, we estimated the capacity of a communication network for Cloud XR video stream and found an optimal Cloud XR video bitrate adaptation function that maximizes the capacity. We considered the capacity as the maximal average XR video bitrate for which the stalling probability is below the pre-defined threshold given the load imposed on the network by the high-priority traffic. Note that because of the considered strict-priority service policy, the high-priority traffic was not affected by the XR video flow. With simulations, we demonstrated the accuracy of the model in estimating the target QoE metrics in the relevant range of scenarios. Finally, the simulations showed that, in the considered scenario, the optimal bitrate adaptation function provides up to 2 times higher average bitrate than one of the state-of-the-art while keeping the stalling probability below the required constraint.

We see multiple possible extensions and applications of the developed model. First, the model can be generalized to a more realistic and complicated high-priority traffic pattern. Second, the model can be extended to take into account the peculiarities of the video encoding: different types of frames in the video stream, their sizes, and their impact on the QoE. Third, the model can be extended to take into account the interference from the other video flows. Finally, a more advanced optimization technique can be applied to reduce the computational complexity of finding the optimal bitrate adaptation function for the particular network state. With such a technique, we can perform optimization with a greater granularity and provide higher QoE to the end-users.

References

[1] E. D. Innocenti, M. Geronazzo, D. Vescovi, R. Nordahl, S. Serafin, L. A. Ludovico, and F. Avanzini, "Mobile virtual reality for musical genre learning in primary education," Computers & Education, vol. 139, pp. 102 – 117, 2019.

[2] T. Joda, G. Gallucci, D. Wismeijer, and N. Zitzmann, "Augmented and virtual reality in dental medicine: A systematic review," Computers in biology and medicine, vol. 108, pp. 93–100, 2019.

[3] R. Oberhauser and C. Pogolski, "Vr-ea: Virtual reality visualization of enterprise architecture models with archimate and bpmn," in International Symposium on Business Modeling and Software Design, pp. 170–187, Springer, 2019.

[4] Z. Lv, X. Li, B. Zhang, W. Wang, Y. Zhu, J. Hu, and S. Feng, "Managing big city information based on webvrgis," IEEE Access, vol. 4, pp. 407–415, 2016.

[5] X. Li, Z. Lv, W. Wang, B. Zhang, J. Hu, L. Yin, and S. Feng, "Webvrgis based traffic analysis and visualization system," Advances in Engineering Software, vol. 93, pp. 1–8, 2016.

[6] J. Wolfartsberger, "Analyzing the potential of virtual reality for engineering design review," Automation in Construction, vol. 104, pp. 27 – 37, 2019.

[7] V. Angelov, E. Petkov, G. Shipkovenski, and T. Kalushkov, "Modern virtual reality headsets," in 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), pp. 1–5, IEEE, 2020.

[8] H. Zhang, J. Zhang, X. Yin, K. Zhou, and Z. Pan, "Cloud-to-end rendering and storage management for virtual reality in experimental education," Virtual Reality & Intelligent Hardware, vol. 2, no. 4, pp. 368–380, 2020.

[9] R. Yadav, W. Zhang, O. Kaiwartya, P. R. Singh, I. A. Elgendy, and Y.-C. Tian, "Adaptive energy-aware algorithms for minimizing energy consumption and sla violation in cloud computing," IEEE Access, vol. 6, pp. 55923–55936, 2018.

[10] W. Zhang, J. Chen, Y. Zhang, and D. Raychaudhuri, "Towards efficient edge cloud augmentation for virtual reality mmogs," in Proceedings of the Second ACM/IEEE Symposium on Edge Computing, pp. 1–14, 2017.

[11] ZTE, "5G Cloud XR application white paper," tech. rep., September 2019.

[12] Huawei, "Huawei helps china mobile fujian release world's first operator cloud vr." https://www.huawei.com/en/news/2018/7/ChinaMobile-Fujian-Operator-Cloud-VR, 2018. Accessed: 06-11-2020.

[13] Vodafone, "5g standalone takes virtual reality teaching to the next level." https://newscentre.vodafone.co.uk/features/5g-standalone-takes-virtual-reality-teaching-to-the-next-level/, 2020. Accessed: 06-11-2020.

[14] Huawei, "Huawei and telefónica jointly demonstrate the industry's first 5g slicing-based interactive vr service." https://www.huawei.com/ch-en/news/2018/2/5g-slicing-based-interactive-vr-service, 2018. Accessed: 06-11-2020.

[15] F. Hu, Y. Deng, W. Saad, M. Bennis, and A. H. Aghvami, "Cellular-connected wireless virtual reality: Requirements, challenges, and solutions," IEEE Communications Magazine, vol. 58, no. 5, pp. 105–111, 2020.

[16] Huawei, "White paper for 5G Cloud VR service experience standards," tech. rep., June 2019.

[17] S. Ahsan, A. Hourunranta, I. D. Curcio, and E. Aksu, "Frisbe: adaptive bit rate streaming of immersive tiled video," in Proceedings of the 25th ACM Workshop on

Packet Video, pp. 28–34, 2020.

[18] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen, "Tiling in interactive panoramic video: Approaches and evaluation," IEEE Transactions on Multimedia, vol. 18, no. 9, pp. 1819–1831, 2016.

[19] MPEG, "ISO/IEC iso/iec 23090-2:2019 MPEG-I Coded Representation of Immersive Media – Part 2: Omnidirectional Media Format," tech. rep., ISO.

[20] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," IEEE Transactions on Communications, vol. 66, no. 11, pp. 5621–5635, 2018.

[21] S. Zhang, M. Tao, and Z. Chen, "Exploiting caching and prediction to promote user experience for a real-time wireless vr service," in 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6, 2019.

[22] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," IEEE Communications Magazine, vol. 55, no. 6, pp. 110–117, 2017.

[23] R. Yadav, W. Zhang, H. Chen, and T. Guo, "Mums: Energy-aware vm selection scheme for cloud data center," in 2017 28th International Workshop on Database and Expert Systems Applications (DEXA), pp. 132–136, IEEE, 2017.

[24] R. Yadav and W. Zhang, "Mereg: Managing energy-sla tradeoff for green mobile cloud computing," Wireless Communications and Mobile Computing, vol. 2017, 2017.

[25] R. Yadav, W. Zhang, K. Li, C. Liu, M. Shafiq, and N. K. Karn, "An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center," Wireless Networks, vol. 26, no. 3, pp. 1905–1919, 2020.

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

[27] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Edge computing meets millimeter-wave enabled vr: Paving the way to cutting the cord," in 2018 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6, IEEE, 2018.

[28] T. Xu, Y. Sun, S. Xia, H. Li, L. Luo, and Z. Chen, "Optimal bandwidth allocation with edge computing for wireless vr delivery," in 2019 IEEE/CIC International Conference on Communications in China (ICCC), pp. 903–907, IEEE, 2019.

[29] "Webxr device api w3c working draft." https://www.w3.org/TR/webxr/. Accessed: 19-01-2021.

[30] Z. Lv, T. Yin, X. Zhang, H. Song, and G. Chen, "Virtual reality smart city based on webvrgis," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 1015–1024, 2016.

[31] Z. Lv, X. Li, H. Lv, and W. Xiu, "Bim big data storage in webvrgis," IEEE Transactions on Industrial Informatics, vol. 16, no. 4, pp. 2566–2573, 2020.

[32] N. Barman and M. G. Martini, "QoE modeling for HTTP adaptive video streaming–a survey and open challenges," IEEE Access, vol. 7, pp. 30831–30859, 2019.

[33] R. McCool, K. A. Abdel, Rahman, and G. Somadder, "Stadia streaming tech: A deep dive." https://www.youtube.com/watch?v=9Htdhz6Op1I, 2019. Accessed: 18-09-2020.

[34] M. Suznjevic, I. Slivar, and L. Skorin-Kapov, "Analysis and qoe evaluation of cloud gaming service adaptation under different network conditions: The case of nvidia geforce now," in 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6, IEEE, 2016.

[35] J. Xu, B. Zhou, C. Zhang, N. Ke, W. Jin, and S. Hao, "The impact of bitrate and GoP pattern on the video quality of H.265/HEVC compression standard," in 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 1–5, 2018.

[36] I. F. Akyildiz, E. Khorov, A. Kiryanov, D. Kovkov, A. Krasilov, M. Liubogoshchev, D. Shmelkin, and S. Tang, "xStream: A new platform enabling communication between applications and the 5G network," in 2018 IEEE Globecom Workshops (GC Wkshps), pp. 1–6, IEEE, 2018.

[37] F. Pollaczek, "Über eine aufgabe der wahrscheinlichkeitstheorie. i," Mathematische Zeitschrift, vol. 32, no. 1, pp. 64–100, 1930.

[38] H. Tijms, "Heuristics for finite-buffer queues," Probability in the Engineering and Informational Sciences, vol. 6, no. 3, pp. 277–285, 1992.

[39] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over http," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 562–585, 2018.

[40] ISO, ISO/IEC 23009-5:2017 Information technology – Dynamic adaptive streaming over HTTP(DASH) – Part 5: Server and network assisted DASH (SAND). ISO.

[41] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, pp. 1–9, IEEE, 2016.

[42] DASH-IF, "dash.js." https://github.com/Dash-Industry-Forum/dash.js. Accessed: 18-09-2020.

[43] K. Spiteri, R. Sitaraman, and D. Sparacio, "From theory to practice: Improving bitrate adaptation in the DASH reference player," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 15, no. 2s, pp. 1–29, 2019.

[44] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adapta-

tion: Evidence from a large video streaming service," in Proceedings of the 2014 ACM conference on SIGCOMM, pp. 187–198, 2014.
[45] Y. Qin, R. Jin, S. Hao, K. R. Pattipati, F. Qian, S. Sen, B. Wang, and C. Yue, "A control theoretic approach to abr video streaming: A fresh look at PID-based rate adaptation," in IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp. 1–9, 2017.
[46] S. Chen, Y. Zhang, H. Peng, and J. Yan, "A joint bitrate and buffer control scheme for low-latency live streaming," in International Conference on Intelligent Science and Big Data Engineering, pp. 369–380, Springer, 2019.
[47] X. Xiao, W. Wang, T. Chen, Y. Cao, T. Jiang, and Q. Zhang, "Sensor-augmented neural adaptive bitrate video streaming on UAVs," IEEE Transactions on Multimedia, vol. 22, no. 6, pp. 1567–1576, 2020.
[48] S. Tian, M. Yang, and W. Zhang, "A practical low latency system for cloud-based vr applications," in International Conference on Communications and Networking in China, pp. 73–81, Springer, 2019.
[49] N. K. Jaiswal, Priority Queues. Elsevier, 1968.
[50] N. Zhirnov, A. Lyakhov, and E. Khorov, "Mathematical model of a network slicing approach for video and web traffic," Journal of Communications Technology and Electronics, vol. 64, no. 8, pp. 890–899, 2019.
[51] L. Takács, "A single-server queue with poisson input," Operations Research, vol. 10, no. 3, pp. 388–394, 1962.
[52] V. Benes, "On queues with poisson arrivals," The Annals of Mathematical Statistics, pp. 670–677, 1957.
[53] L. Takács et al., "The transient behavior of a single server queuing process with a poisson input," in Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Contributions to Probability Theory, The Regents of the University of California, 1961.
[54] J. Abate and W. Whitt, "Transient behavior of the M/G/1 workload process," Operations Research, vol. 42, no. 4, pp. 750–764, 1994.
[55] L. Takács, "Occupation time problems in the theory of queues," in Mathematical Methods in Queueing Theory, pp. 91–131, Springer, 1974.
[56] L. Kleinrock, "Queueing systems. volume i: Theory," 1975.
[57] A. Lyakhov, D. Ostrovsky, and E. Khorov, "Analytical study of the quality of links established by the neighborhood discovery protocol," Journal of Communications Technology and Electronics, vol. 57, no. 12, pp. 1314–1321, 2012.
[58] "The network simulator ns-3." https://www.nsnam.org. Accessed: 18-09-2020.
[59] "TGax Simulation Scenarios." https://mentor.ieee.org/802.11/dcn/14/11-14-0980-16-00ax-simulation-scenarios.docx, Nov. 2015. Accessed: 13-11-2020.
[60] W. J. Stewart, Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling. Princeton university press, 2009.
[61] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," Proceedings of the IEEE, vol. 106, no. 10, pp. 1834–1853, 2018.

MIKHAIL LIUBOGOSHCHEV (M'2019) received his B.S. and M.S. degrees in applied mathematics and physics from Moscow Institute of Physics and Technology, Moscow, Russia, in 2017 and 2019, respectively. He is currently pursuing a Ph.D. degree in telecommunications at Moscow Institute of Physics and Technology, Moscow, Russia. He is a researcher at Network Protocols Research Laboratory and Wireless Networks Lab (both Kharkevich Institute for Information Transmission, Russian Academy of Sciences) since 2016 and 2018, respectively. His research interests are to 5G and beyond wireless systems, QoS-aware cross-layer optimization, and stochastic network modeling and optimization. He participates in national and international projects and does research within the framework of joint research projects with the leading telecommunication companies.

KAMILLA RAGIMOVA was born in Dubna, Russia. She received the B.Sc. degree in applied mathematics and physics from Moscow Institute of Physics and Technology, Moscow, Russia, in 2020. Currently, she is an M.Sc. student at Higher School of Economics, Moscow, Russia. From 2018 to 2020, she was an intern at Wireless Networks Lab, Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences. Her research interests include the analysis of the video-on-demand streaming services, modeling of adaptive virtual reality applications.

ANDREY LYAKHOV is a Full Professor, the Deputy Director, and the Head of the Network Protocols Research Laboratory, Institute for Information Transmission Problems, Russian Academy of Sciences. His main research interests are related to the design and analysis of wireless network protocols, wireless network performance evaluation methods, and stochastic modeling of wireless networks based on random multiple access. He has over 20 years of experience in Wi-Fi networks design and performance evaluation. He has authored three monographs, more than 100 papers cited in Scopus, and has ten patents. He was a member of technical and program committees of large IT conferences (ICC, MACOM, MobiHoc, Networking, and MASS) and a general chair of IEEE BlackSeaCom 2019 and WiFlex 2013. He was a recipient of many international and Russian awards. He led many joint research projects with top telecommunication companies and collaborative projects (e.g., FP7 ICT collaborative project FLAVIA "Flexible Architecture for Virtualizable wireless future Internet Access," from 2010 to 2012).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3062555, IEEE Access

SIYU TANG received her M.Sc. and Ph.D. degree in Electrical Engineering from Delft University of Technology, The Netherlands, in 2006 and 2010 respectively. Since then, she was with Bell Labs, Alcatel-Lucent (later merged with Nokia), Antwerp, Belgium, working on novel algorithms and network protocols for ultra-low latency networks. From 2017, she joined Huawei Munich Research Center, Germany, as a principal researcher, working in the field of Telecommunications networks (e.g., future Internet architecture, next generation network protocols) and Industrial Communication Networks (e.g., time sensitive networking, DetNet). Her expertise is to apply queuing theories, stochastic modeling methodologies and control theories in communication networks to improve their performance, stability and connectivity.

EVGENY KHOROV (Senior Member, IEEE) is the Head of the Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences. He has led dozens of national and international projects sponsored by academic funds and industry. Being a voting member of IEEE 802.11, he has contributed to the 802.11ax standard as well as to the Real-Time Applications TIG with many proposals. He has authored more than 100 articles. His main research interests are related to 5G and beyond wireless systems, next-generation Wi-Fi, protocol design, and QoS-aware cross-layer optimization. He was a recipient of the Russian Government Award in Science, several Best Papers Awards, and Scopus Award Russia 2018. In 2015, 2017, and 2018, Huawei RRC awarded him as the Best Cooperation Project Leader. He gives tutorials and participates in panels at large IEEE events. He chairs TPC of the IEEE GLOBECOM 2018 CA5GS Workshop and IEEE BLACKSEACOM 2019. He was awarded as the Editor of the Year 2020 of *Ad Hoc Networks*.