

Quality of Experience Oriented Cross-layer Optimization for Real-time XR Video Transmission

Guangjin Pan, Shugong Xu, *Fellow, IEEE*, Shunqing Zhang, *Senior Member, IEEE*,
Xiaojing Chen, *Member, IEEE*, and Yanzan Sun, *Member, IEEE*

Abstract—Extended reality (XR) is one of the most important applications of beyond 5G and 6G networks. Real-time XR video transmission presents challenges in terms of data rate and delay. In particular, the frame-by-frame transmission mode of XR video makes real-time XR video very sensitive to dynamic network environments. To improve the users' quality of experience (QoE), we design a cross-layer transmission framework for real-time XR video. The proposed framework allows the simple information exchange between the base station (BS) and the XR server, which assists in adaptive bitrate and wireless resource scheduling. We utilize the cross-layer information to formulate the problem of maximizing user QoE by finding the optimal scheduling and bitrate adjustment strategies. To address the issue of mismatched time scales between two strategies, we decouple the original problem and solve them individually using a multi-agent-based approach. Specifically, we propose the multi-step Deep Q-network (MS-DQN) algorithm to obtain a frame-priority-based wireless resource scheduling strategy and then propose the Transformer-based Proximal Policy Optimization (TPPO) algorithm for video bitrate adaptation. The experimental results show that the TPPO+MS-DQN algorithm proposed in this study can improve the QoE by 3.6% to 37.8%. More specifically, the proposed MS-DQN algorithm enhances the transmission quality by 49.9%-80.2%.

Index Terms—Wireless extended reality, adaptive bitrate, QoE, reinforcement learning, cross-layer design.

I. INTRODUCTION

RECENTLY, Metaverse [1] sparks people's imagination about immersive experiences. As a result, extended reality (XR) becomes one of the most critical 5G media applications under deliberation, because XR can provide people with immersive experiences. XR encompasses various types of reality, including Augmented Reality (AR), Virtual Reality (VR), Mixed Reality (MR) and Cloud Gaming (CG) [2], [3], all of which require low-latency transmission and interaction. It poses new challenges for beyond 5G and 6G networks.

The use of XR in amalgamating virtual and real worlds presents users with an immersive experience that requires significant computing power. Hence, XR services are generally rendered on cloud or edge servers and then transmitted to

the XR clients for playback. XR services fall under the category of Real-Time Broadband Communication (RTBC) services, which comprise enhanced mobile broadband (eMBB) communication and ultra-reliable low latency communication (URLLC) requirements. To ensure a seamless, high-quality XR video experience, the system must fulfill the standards of low latency and high data rate. Unfortunately, the existing adaptive real-time XR video streaming systems over wireless networks pose two main challenges.

- *The challenge of bitrate control for XR servers.* For XR servers, the challenge is to adjust the bitrate rapidly and accurately so as to adapt to the network environment. In traditional chunk-based video adaptive wireless streaming systems, users combine historical bandwidth and buffer status to adjust the bitrate adaptively [4]. Updates to bandwidth estimation and buffer status typically take seconds. However, second-level bandwidth estimation cannot meet frame-level transmission requirements in real-time XR systems. Due to the frame-by-frame transmission and playback mode of the real-time XR video, even slight network jitter can impact the performance of XR transmission. To overcome this challenge, the bitrate of XR videos can be adjusted based on the user feedback information, such as loss rate and packet arrival delay gradient [5]. However, this information can only indirectly reflect the network status, and the system lacks lower-layer network observations for bitrate adjustment.
- *The Challenge of resource scheduling for the base station (BS).* 5G has larger transmission bandwidth and lower latency to meet the requirements of eMBB and URLLC scenarios. The real-time XR transmission requires both low latency and high data rates, which poses new challenges for wireless systems. Additionally, different XR video frames may have varying levels of importance. For example, in the Group of Picture (GOP)-based XR video encoding model [6], I-frames are usually more important than P-frames. Therefore, if wireless resources are limited, the scheduler should prioritize transmitting important information. However, traditional scheduling policies based on traffic flow priority, such as [7] and [8], can not handle this problem well. Meanwhile, the dynamic wireless environment and resources make real-time XR video transmission more difficult.

These challenges motivate our study on a new cross-layer transmission architecture for real-time XR video. We consider a real-time XR oriented wireless transmission system. As

G. Pan, S. Xu, S. Zhang, X. Chen and Y. Sun are with School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China. Emails: {guangjin_pan, shugong, shunqing, jodiechen, yanzansun}@shu.edu.cn.

Part of this work has been accepted by FCN-2023. This work was supported in part by the National Key R&D Program of China under Grant 2022YFB2902005, the National High Quality Program grant TC220H07D, the National Natural Science Foundation of China (NSFC) under Grant 61871262, 62071284, and 61901251, the National Key R&D Program of China grants 2022YFB2902000, Foshan Science and Technology Innovation Team Project grant FS0AAKJ919-4402-0060. The corresponding author is Shugong Xu.

shown in Fig. 1, the edge XR server performs real-time video encoding and transmits each encoded video frame to BS with each frame consisting of a set of video packets. Subsequently, the BS schedules the video packets in the queue and delivers them to the respective XR users, who then decode and play the video. The contributions of our study are listed below,

- *A cross-layer transmission framework for real-time XR video.* We propose a framework for real-time XR video transmission. The proposed framework enables the XR server and BS to assist each other in improving the quality of experience (QoE) for XR videos. Based on the proposed framework, we model the problem as a cross-layer optimization problem between the Medium Access Control (MAC) and application layers, intending to improve the QoE of real-time XR video. In order to address the problem of mismatch between the wireless resource allocation and adaptive bitrate at two time-scales, we decouple the problem into two problems and then solve each separately.
- *A frame-priority-based wireless resource scheduling scheme at the MAC layer.* At the MAC layer, we propose a frame-priority based scheduling scheme to maximize the frame success rate. The resource allocation for each time slot involves a highly complex and large action space, as it requires the allocation of a variable number of resource blocks (RBs) for each video frame. To solve this problem, we propose a multi-step Deep Q-network (MS-DQN) algorithm to reduce the action space, allowing the agent to make multi-step decisions within each time slot to allocate wireless resources.
- *A TPPO-based adaptive bitrate scheme at the application layer.* At the application layer, due to the XR server's inability to obtain global information, the problem is modelled as a partially observable Markov decision process (POMDP). Therefore, we need a method that can extract the underlying relationships between video bitrate selection and communication modes from historical observation sequences to choose the appropriate video bitrate. We introduce a Transformer-based Proximal Policy Optimization (TPPO) algorithm to extract semantic information from the historical environment and intelligently select the optimal video bitrate to improve the QoE of real-time XR video.

Compared with the baseline algorithms Proportional Fair (PF) [9], PF-I [10], and Deep Deterministic Policy Gradient (DDPG) [8], the proposed MS-DQN explores and learns the optimal scheduling strategy through interaction with the environment. At the same time, MS-DQN reduces the action space of each step decision by decomposing the single-step decision process into a multi-step decision, thus reducing the learning difficulty. In comparison with baseline algorithms DDPG [11] and Proximal Policy Optimization (PPO) [12], the proposed bitrate adjustment algorithm TPPO combines the advantages of the PPO algorithm and transformer to extract semantic information from the historical environment. The numerical results show that our proposed TPPO+MS-DQN algorithm can outperform the baseline algorithms. The MS-

DQN-based algorithm can increase the transmission quality by 49.9%-80.2%, and the proposed TPPO+MS-DQN algorithm can improve QoE by 3.6%-37.8%.

The rest of this paper is organized as follows. Sec. II provides a detailed overview of related works. In Sec. III, we introduce system models, and in Sec. III, we formulate the problem. In Sec. V, we describe the proposed MS-DQN algorithm. Sec. VI gives the proposed TPPO algorithm. Numerical results and analysis are presented in Sec. VII. Finally, the paper is concluded in Section VIII.

II. RELATED WORK

Several related works have been developed on wireless scheduling, as discussed in [7]–[10], [13], [14]. The proportional fair scheduler was analyzed in [9] to obtain the cell throughput, while scheduling algorithms like Exponential Rule (EXP-RULE) [13] and Modified Earliest Deadline First and Proportion Fair (M-EDF-PF) [7] were proposed to support real-time traffics. In [8], the DDPG algorithm was utilized to minimize queuing delay experienced by users. Additionally, a delay-oriented scheduling algorithm based on partially observable Markov decision process was proposed in [14] to reduce the tail delay and average delay. However, these works do not consider the traffic characteristics of XR. [10] focused on frame-level integrated transmission to enhance the frame success rate.

Adaptive bitrate algorithms have been extensively researched for chunk-based video-on-demand transmission. [4], [12], [15]–[20] use bandwidth prediction, video buffer size, or both to adjust bitrates of video chunks to improve the QoE. [15] uses a control-theoretic method for dynamic adaptive bitrate streaming, and proposes a model predictive control algorithm. Pensieve [4] uses the Deep Reinforcement Learning (DRL) algorithm to learn and make adaptive bitrate (ABR) decisions solely by observing the result performance of past decisions. BOLA [16] uses Lyapunov optimization to minimize rebuffering and maximize video quality. Fleet [17] considers the idle period of the Live video streaming, and designs a HyperText Transfer Protocol (HTTP) chunk-level measurement algorithm and a stochastic model predictive control (MPC) controller for bitrate control. These works only consider the QoE of a single user, and cannot be applicable to scenarios where multiple video streams compete for network resources. The authors in [18] believe when multiple clients compete at a network bottleneck, it is difficult for a client to correctly perceive its fair-share bandwidth. They propose the PANDA [18] algorithm to probe the fair-share bandwidth. The authors in [19] propose the MUABR algorithm based on multi-agent deep reinforcement learning, which combines ABR algorithm with bandwidth allocation strategy to optimize the overall QoE and can improve viewing experience of each user as much as possible. The authors in [20] employ Lyapunov optimization techniques for bandwidth allocation among multiple users. In [12], the authors further enhanced the average QoE of users using a PPO-based ABR algorithm. The aforementioned schemes address the problem of bitrate adaptation through the perspective of end-to-end optimization but do not consider the role of wireless systems.

In wireless video transmission systems, reasonable resource allocation in the BS helps improve the quality of video transmission and ensures fairness among users [21]. Correspondingly, the lower layer information in the BS is also helpful in adjusting the video bitrate of users [22]. Therefore, a large amount of research has focused on the cross-layer optimization in wireless video transmission systems [11], [21]–[25]. [22] combines higher and lower layers of information and proposes a QoE-oriented adaptive video streaming algorithm based on dueling DQN. In multiple input multiple output (MIMO) systems, authors in [21] jointly optimize the beamforming vector and video bitrate, and propose an optimization-based beamforming scheme and a DRL-based bitrate adaptation scheme. In orthogonal frequency division multiple (OFDM) systems, [23] jointly optimize communication resources, transmit power, and the video bitrate. They derive an online optimization problem to approximate the offline problem, and solve the online problem by decomposing it into wireless resource allocation problem and bitrate adaptation problem. In [24], the authors propose a non-orthogonal multiple access (NOMA) based uncoded multiuser video transmission system. They optimize the allocation of power and channel resources to guarantee a high-quality linear video delivery. The authors in [25] aim to design an online joint transcoding and transmission resource allocation algorithm to maximize the users' QoE. In [11], the authors jointly design caching, computing, and bitrate adaption to improve user's QoE. However, the above works only consider the transmission and bitrate adaption of chunk-based video and cannot be applied to the frame-based real-time XR video transmission. In [26], the authors consider the uplink multi-user video transmission in an OFDM system. They utilize channel state information and rate distortion (RD) information for subcarrier allocation and power allocation, aiming to optimize the sum of distortions at each time slot. Paper [27] further investigates the cross-layer resource allocation problem in a multi-antenna BF-NOMA-OFDMA system, with the goal of reducing the sum of MSE distortions in video transmission. The authors in [28] consider multi-user video communication on the uplink OFDMA system and a cross-layer algorithm is proposed for joint bit allocation, packet scheduling and wireless resource assignment are proposed to minimize the end-to-end expected video distortion. However, these existing works [26]–[28] do not consider the problem of mismatch between bitrate adaption and resource allocation at two time-scales. In addition, these studies model the quality of videos using the RD function at the level of GoP, which cannot be applied to real-time XR video transmission.

For real-time XR video transmission, Google propose the Google Congestion Control (GCC) congestion control algorithm for the Web real-time communication (WebRTC) system [5]. [29] uses reinforcement learning and federated learning to optimize real-time video transmission. However, these end-to-end adaptive bitrate methods do not consider the assistance of wireless systems for video transmission. In [30], a cross-layer resource allocation technique is proposed for wireless video transmission along with each forward error correction (FEC) block. The authors in [31] consider a joint video packet assignment, power control and user scheduling problem to

maximize the real-time video transmission quality for NOMA networks. However, they do not consider the bitrate adaptation.

III. SYSTEM MODEL

In this section, we consider an XR video 5G transmission system with one edge XR server, one BS, and N XR users denoted by the set $\mathcal{N} = \{1, 2, \dots, N\}$. The time is assumed to be discretized into several slots $\mathcal{T} = \{0, 1, \dots, t, \dots\}$. The length of one slot Δt is equal to the time length of one transmission time interval (TTI) of the 5G system. Therefore, the BS allocates RBs once in each time slot t .

Fig. 1 shows our proposed framework for edge XR streaming. In this framework, the BS schedules XR video packets using a frame-priority-based scheduling scheme based on available frame information. The edge XR server monitors the network state of the BS to adjust the video bitrate dynamically. Specifically, in the edge XR server, the bitrate adjustment module includes A-① Video Generator, A-② Radio Access Network (RAN) Monitor, A-③ XR Bitrate Controller, A-④ XR Video Transcoder, and A-⑤ Real-Time Transport Protocol (RTP) Sender. The A-① Video Generator generates XR videos in real time. The A-② RAN Monitor collects the network information from the BS, including average rate, bandwidth, frame transmission success rate, and RB Utilization. The ③ XR Bitrate Controller selects an appropriate bitrate based on the collected network information, and the A-④ XR Video Transcoder encodes the video stream accordingly. The A-⑤ RTP sender then packets the video and marks RTP header information before transmitting it to the BS via several IP packets. On the BS, the frame-based scheduling pool is constructed by the B-① Scheduling Pool to identify and schedule packets according to the frame information from the RTP header. The B-② Resource Scheduler in turn schedules packets based on network information. The B-③ Network Monitor periodically sends network information to the XR server to support dynamic bitrate adjustment.

A. XR Traffic Model

Considering the inherent timescale mismatch between the MAC layer and the application layer, we denote $T_b \Delta t$ as the duration of the bitrate adjustment period, and define that the bitrate adjusts in time slot set $\mathcal{T}_b = \{0, T_b, 2T_b, \dots, t_b T_b, \dots\} \subset \mathcal{T}$. For convenience, we define t_b as the bitrate adjustment period in time slot $t_b T_b$.

In the XR server, video frames are generated at regular intervals based on a predefined frame rate. One encoding scheme is GOP-based, where I-frames are transmitted periodically every K frames, and the remaining frames are P-frames [6]. Let $\mathcal{K}_n$ denote the frame set of user n . $\mathcal{K}_{n,t_b} \subset \mathcal{K}_n$ is the set of frames of user n within the bitrate adjustment period t_b . The frame rate of the user's XR video is denoted as f_n , and α represents the average ratio of size between one I-frame and one P-frame [6]. In the time slot $t_b T_b \leq t < (t_b + 1)T_b$, the video bitrate of user n is R_{n,t_b} , and the frame size of the k -th ($k \in \mathcal{K}_{n,t_b}$)

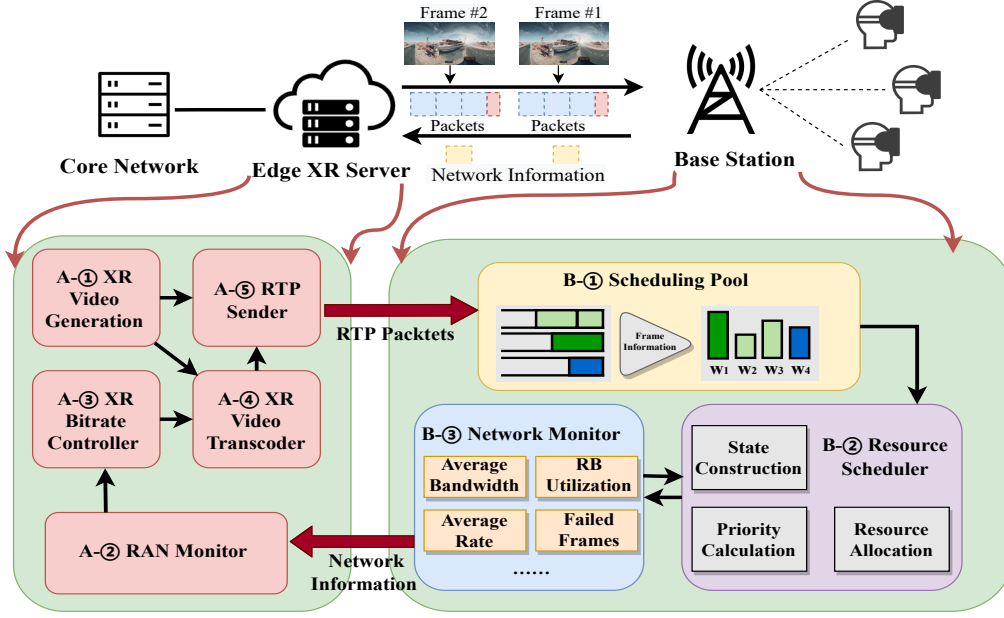


Fig. 1. XR transmission system framework. Real-time XR video is rendered on edge XR servers and then transmitted to clients for playback. BS uses frame information to schedule packets and improve transmission quality.

video frame arriving is represented by $r_{n,k}$. As reported in [6], the average size of $r_{n,k}$ is,

$$\mathbb{E}[r_{n,k}] = \begin{cases} \frac{R_{n,t_b}}{f_n} \cdot \frac{K\alpha}{K-1+\alpha}, & \text{if } d_{n,k} = 1, \\ \frac{R_{n,t_b}}{f_n} \cdot \frac{K}{K-1+\alpha}, & \text{if } d_{n,k} = 0, \end{cases}$$

where $d_{n,k}$ is used to indicate the type of frame, with a value of either 0 or 1. Specifically, if $d_{n,k} = 1$, the frame is an I-frame, while a value of 0 corresponds to a P-frame.

The k -th frame's arrival time of user n can be expressed as,

$$t_{n,k} = t_{n,0} + \left\lfloor \frac{\frac{k-1}{f_n} + t_{jitter}}{\Delta t} \right\rfloor, \quad (1)$$

where $t_{n,0}$ is the initial arrival time, f_n is the frame rate, and t_{jitter} is the arrival jitter time due to the codec and transmission. The value of t_{jitter} is a random variable that follows a certain distribution.

In addition, considering the different types of frames and encoding methods, different XR video frames have varying levels of importance. For example, I-frames contain all the information of a video frame and can be decoded independently, while P-frames require the information from the previous frame for decoding. Therefore, compared to P-frames, I-frames have more data and are more important. For each user n , we define $w_{n,k}$ as the importance weight for frame k ¹.

B. Channel and Scheduling Models

Considering there are N_t^{rb} available RBs in time slot t , for each RB, the achievable data rate of user n at the t -th slot is

¹The optimal design of the importance weight depends on the type of video encoder and the content of the video. In this paper, we mainly discuss how to design scheduling algorithms to optimize real-time XR video transmission in the presence of frame importance weight. The optimization of importance weights is not our primary focus in this discussion.

given by,

$$c_{n,t} = \Delta t B_w \log(1 + \frac{p_n h_{n,t}}{B_w \sigma^2}), \text{ bits/s} \quad (2)$$

where p_n is the transmit power of user n , $h_{n,t}$ is the channel gain of user n in time slot t , B_w is the bandwidth of each RB, σ^2 is the Gaussian noise power spectral density. $c_{n,t}$ represents the amount of data that each RB can transmit.

In the XR video transmission task, it is essential to fully transmit packets belonging to the same frame before the frame delay budget (FDB) t^* [10]. Let $\tilde{r}_{n,k,t}$ denote the remaining data size of frame k waiting for transmission in time slot t . It can be given by,

$$\tilde{r}_{n,k,t} = r_{n,k} - \sum_{\tau=t_{n,k}}^{t-1} N_{n,k,\tau}^{rb} c_{n,\tau}, \quad (3)$$

where $r_{n,k}$ represents the total size of the frame, $N_{n,k,t}^{rb}$ denotes the number of RBs allocated to user n for frame k in time slot t , and $c_{n,\tau}$ represents the achievable data rate of user n in RBs at time slot τ , as calculated in Equation (2).

Let $x_{n,k}$ represent the indicator of whether the video frame is successfully transmitted. Therefore, a frame can be considered as successfully transmitted if the condition

$$x_{n,k} = \mathbb{I}(\tilde{r}_{n,k,t_{n,k}+t^*} \leq 0) = 1 \quad (4)$$

is satisfied, which indicates that the allocated wireless resources are capable of transmitting a larger amount of data than the size of the video frames. Otherwise, $x_{n,k} = 0$.

To enhance the transmission capability of XR in wireless networks, we propose that the BS extracts some frame infor-

mation from the real-time transport protocol (RTP) header². By utilizing this information, the BS can sort the packets waiting to be transmitted and schedule them accordingly. Specifically, the set of frames waiting to be transmitted during time slot t is denoted by $\mathcal{K}_t = \{(n, k) | n \in \mathcal{N}, k \in \mathcal{K}_{n,t}\}$, where $\mathcal{K}_{n,t} = \{k | k \in \mathcal{K}_n, 0 \leq t - t_{n,k} < t^*, \tilde{r}_{n,k,t} > 0\}$ represents the set of frames for user n that are waiting to be transmitted in time slot t .

The BS can perform scheduling based on the priority of frames, which is determined by the set of frames received. Let $p_{n,k,t}$ denote the priority of user n for frame k in time slot t . $N_{n,k,t}^{rb}$ is the number of RBs allocated to each video frame based on frame priority. The resource allocation policy can be expressed as,

$$N_{n,k,t}^{rb} = \begin{cases} \lceil \frac{\tilde{r}_{n,k,t}}{c_{n,t}} \rceil, & \text{if } \tilde{r}_{n,k,t} < c_{n,t} (N_t^{rb} - \sum_{(n',k') \in \mathcal{K}_{n,k,t}} N_{n',k',t}^{rb}), \\ N_t^{rb} - \sum_{(n',k') \in \mathcal{K}_{n,k,t}} N_{n',k',t}^{rb}, & \text{otherwise,} \end{cases} \quad (5)$$

where $\mathcal{K}_{n,k,t} = \{(n', k') | (n', k') \in \mathcal{K}_t, p_{n,k,t} < p_{n',k',t}\}$ is the set of frames with a priority higher than the frame (n, k) . The resource allocation policy indicates that, given the priority of video frames, the resource scheduler will prioritize allocating sufficient wireless resources to higher-priority video frames.

The traditional flow-priority-based scheduling policy primarily considers the user's queue lengths (to ensure fairness) and the channel state information (to optimize throughput) [14]. In the frame-priority-based scheduler, the scheduler utilizes frame information and channel state information for scheduling. The priority is calculated by,

$$\mathbf{p}_t = \pi^p(\mathbf{r}_t, \tilde{\mathbf{r}}_t, \boldsymbol{\tau}_t, \mathbf{w}_t, \mathbf{h}_t), \quad (6)$$

where $\mathbf{p}_t \triangleq \{p_{n,k,t}, \forall (n, k) \in \mathcal{K}_t\}$, $\mathbf{r}_t \triangleq \{r_{n,k,t}, \forall (n, k) \in \mathcal{K}_t\}$, $\tilde{\mathbf{r}}_t \triangleq \{\tilde{r}_{n,k,t}, \forall (n, k) \in \mathcal{K}_t\}$, $\boldsymbol{\tau}_t \triangleq \{\tau_{n,k,t}, \forall (n, k) \in \mathcal{K}_t\}$, $\mathbf{w}_t \triangleq \{w_{n,k,t}, \forall (n, k) \in \mathcal{K}_t\}$, and $\mathbf{h}_t \triangleq \{h_{n,t}, \forall (n) \in \mathcal{N}\}$. Specifically, $\tau_{n,k,t} = t^* + t_{n,k} - t$ is the remaining FDB (RFDB). $\pi^p(\cdot)$ represents the scheduler policy based on video frame information and channel information.

C. Adaptive XR Video Streaming Model

As mentioned earlier, the XR server periodically gathers network information and performs XR video bitrate adaptation. Using the A-③ XR Bitrate Controller module described above, the XR Bitrate Controller utilizes historical observations of network information and selected bitrates to make decisions on video bitrate selection. Specifically, at the beginning of each bitrate adjustment period, the RAN Monitor samples network information $\mathbf{o}_{t_b}$ from the BS. The XR Bitrate Controller then uses the historical network information observation sequence $\mathbf{o}_{t_b}, \dots, \mathbf{o}_{t_b-t_o}$ and historical bitrate selection results $R_{1,t_b-1}, \dots, R_{1,t_b-t_o}, \dots, R_{N,t_b-1}, \dots, R_{N,t_b-t_o}$ to

determine the video bitrate R_{1,t_b} for that adjustment period. t_o represents the length of the observation window. The input to the XR Bitrate Controller can be represented as $\mathbf{O}_{t_b} = \{R_{1,t_b-1}, \dots, R_{1,t_b-t_o}, \dots, R_{N,t_b-1}, \dots, R_{N,t_b-t_o}, \mathbf{o}_{t_b}, \dots, \mathbf{o}_{t_b-t_o}\}$. Therefore, the adaptive bitrate policy of the XR Bitrate Controller can be expressed as

$$R_{n,t_b} = \pi^b(\mathbf{O}_{t_b}), \quad (7)$$

where $\pi^b(\mathbf{O}_{t_b})$ denotes the adaptive bitrate strategy deployed for the XR video.

In chunk-based adaptive video streaming, the QoE of a user is typically defined in terms of video quality, quality variation, and rebuffering time [4]. However, the chunk-based QoE definition cannot directly apply to real-time XR video transmission systems. Inspired by [4], we use a frame-based QoE expression to represent the QoE of adaptive real-time XR video streams. We redefine the QoE of real-time XR video into two parts, i.e., video playback quality $\mathcal{Q}_{n,t_b}^p$ and video transmission quality $\mathcal{Q}_{n,t_b}^t$. The video playback quality is mainly related to the video bitrate, including video quality and quality variations, and the quality of video transmission is related to whether the video frames are successfully transmitted, where

$$\mathcal{Q}_{n,t_b}^p = \beta_1 \frac{R_{n,t_b} |\mathcal{K}_{n,t_b}|}{f_n} - \beta_2 |R_{n,t_b} - R_{n,t_b-1}|, \quad (8)$$

$$\mathcal{Q}_{n,t_b}^t = - \sum_{k \in \tilde{\mathcal{K}}_{n,t_b}} w_{n,k}. \quad (9)$$

β_1 and β_2 are the weight factors, and $\tilde{\mathcal{K}}_{n,t_b}$ is the set of frames that failed to transmit within the FDB. The QoE mainly comprises the following three components,

- *Video Quality*. We define the QoE for video quality as the product of the average frame size and the number of frames transmitted within a bitrate adjustment period, representing the total size of the video sent. It is represented by the first item of (8).
- *Quality Variations*. Similar to [4], the quality variations indicate the magnitude of video quality variation within adjacent bitrate adjustment periods, used to measure the stability of the video bitrate. It is represented by the second item of (8).
- *Weight Sum of Failed Frames*. Transmission failures of frames lead to considerable QoE losses, impacting the user's overall experience negatively. The QoE of transmission is defined as the sum of the importance weight for failed frames during a bitrate adjustment period. We can express the transmission QoE using (9).

IV. PROBLEM FORMULATION

In this paper, our goal is to maximize the QoE of XR video applications. Based on the proposed XR transmission system framework, we aim to find the optimal video bitrate adjustment scheme $\pi^b(\cdot)$ and wireless resource allocation scheme $\pi^p(\cdot)$.

Problem 1 (Original Problem): The original problem can be formulated as the following optimization problem,

$$\underset{\pi^p(\cdot), \pi^b(\cdot)}{\text{maximize}} \quad \sum_{n \in \mathcal{N}} \mathbb{E}[\alpha_p \log \mathcal{Q}_{n,t_b}^p + \alpha_t \mathcal{Q}_{n,t_b}^t] \quad (10)$$

²Although the current RTP standard [32] does not support this assumption, we believe that if the BS can obtain some frame information, the transmission quality of XR video can be effectively improved.

$$\begin{aligned} \text{s.t.} \quad & \sum_{(n,k) \in \mathcal{K}_t} N_{n,k,t}^{rb} \leq N_t, \quad \forall t, \quad (11) \\ & R_{min} \leq R_{n,t_b} \leq R_{max}, \quad \forall t_b, \quad (12) \\ & (5), (6), (7), (8), (9), \end{aligned}$$

where α_p and α_t are weights. R_{max} and R_{min} represent the maximum and minimum XR video bitrate, respectively. In (10), the objective QoE for XR video transmission consists of the video playback quality and the video transmission quality. The video playback quality uses a logarithmic function to improve fairness among users' bitrates. Meanwhile, we use the parameters α_p and α_t to balance the video playback quality and the video transmission quality. (11) constrains the number of RBs in each time slot, and (12) constrains the range of bitrate selection.

The above problem is generally difficult to solve due to the following reasons. Firstly, the problem couples two time-scales. The wireless resource allocation task is performed each slot t and the bitrate selection task is performed each bitrate adjustment period t_b . Secondly, as a long-term optimization problem, the uncertainty of wireless channel state, available network resources, arrival time of frames, frame sizes poses a challenge to effective problem solving. Thirdly, for wireless resource allocation task, it is difficult to obtain an optimal closed-form expression for the priority-based scheduling scheme through mathematical analysis. Finally, the inability of the video server to completely observe the network state brings a challenge for video bitrate adjustments.

Therefore, we propose a multi-agent optimization solution that optimizes wireless resource scheduling and video bitrate allocation separately. The system diagram of the algorithm is shown in Fig. 2. The MS-DQN algorithm utilized on the BS side interacts with the wireless network environment for wireless resource allocation. The XR server employs the TPPO algorithm for the XR video bitrate adaptation. The BS and XR server exchange network and frame information with each other to improve video QoE. Specifically, the proposed MS-DQN is used to optimize the frame transmission success rate based on the frame importance and improve the transmission quality Q_{n,t_b}^t . The TPPO algorithm is used to optimize video bitrate selection, thereby improving both video playback quality Q_{n,t_b}^p and transmission quality Q_{n,t_b}^t .

V. FRAME-PRIORITY-BASED SCHEDULING METHOD IN MAC LAYER

In the MAC layer, the scheduler performs wireless resource scheduling for packets in the scheduling pool. Adopting an appropriate radio resource scheduling scheme can effectively enhance transmission quality. In this section, we describe the frame-priority-based scheduling scheme and formulate the problem to maximize the transmission quality. Then we introduce the proposed MS-DQN algorithm.

A. Frame-Priority based Problem Formulation

We first simplify the *Problem 1* by representing it as a wireless resource allocation problem, and aim to improve the transmission quality.

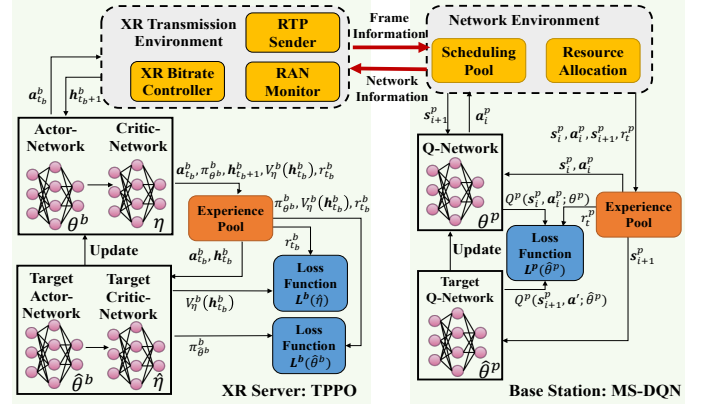


Fig. 2. Algorithmic diagram of the real-time XR transmission system.

Problem 2 (Resource Allocation Subproblem): The wireless resource allocation subproblem can be represented as a scheduling problem based on frame priorities,

$$\begin{aligned} \text{minimize} \quad & \sum_{n \in \mathcal{N}} \mathbb{E}[Q_{n,t_b}^t] \\ \text{s.t.} \quad & (5), (6), (9), (11), \end{aligned} \quad (13)$$

We propose the MS-DQN based resource scheduling framework to solve the problem. When the action space is discrete and finite, DQN can work well. However, in the *Problem 2*, the frame priority $p_{n,k,t}$ is a continuous value, and the action space is infinite. Although we can sort the priorities to select the scheduled users and discretize the action, the action space is still large. To reduce the action space, we proposed a multi-step based DQN algorithm. In our proposed MS-DQN algorithm, in each decision step, the agent outputs the probability values associated with each frame scheduling, which can be regarded as the priority of each frame. The proposed algorithm allows the agent to make multi-step decisions in each slot and select one frame of data to transmit in each decision step. Fig. 3 shows an example of the state transition. In this example, the agent makes a three-step decision in slot t . Then the BS allocates RBs based on actions a_i^p , a_{i+1}^p , and a_{i+2}^p . Then, the state transitions to a_{i+3}^p , which is in slot $t+1$.

B. MDP Reformulation of Problem P2

We define the *Problem 2* as an infinite-horizon Markov decision process (MDP) by defining a tuple with five elements $S^p, \mathcal{A}^p, \mathcal{P}^p, \mathcal{R}^p, \gamma^p$. Here, S^p is the state set, $\mathcal{A}^p$ is the action set, $\mathcal{P}^p$ is the transition probability, $\mathcal{R}^p$ is the reward function, and γ^p is the discount factor ($0 < \gamma^p < 1$). The MDP is defined as follows,

1) **State:** In each decision step i , we assume that the decision step is in time slot t , and $\mathcal{K}_i \subset \mathcal{K}_t$ represents the set of frames waiting for transmission. The state is represented by $s_i^p = \{s_{n,k,i}^p \mid \forall (n,k) \in \mathcal{K}_i\}$, where $s_{n,k,i}^p = (w_{n,k,i}, \tilde{r}_{n,k,i}, \tau_{n,k,i}, c_{n,i}, N_i^{rb})$ consists of the data size waiting for transmission $\tilde{r}_{n,k,i} = \tilde{r}_{n,k,t}$, the RFDB $\tau_{n,k,i} = \tau_{n,k,t}$, the achievable data rate of each RB $c_{n,i} = c_{n,t}$, the number of remaining RBs N_i^{rb} , and the frame importance weight

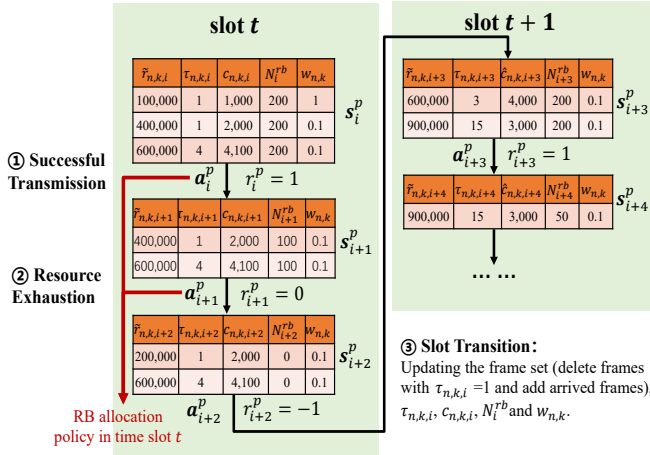


Fig. 3. This example demonstrates three types of state transitions. In slot t , the agent of MS-DQN makes two scheduling decisions and transmits two frames.

$w_{n,k,i} = w_{n,k}$. Therefore, the state s_i^p can be represented as a state matrix of size $(|\mathcal{K}_i|, 5)$.

2) **Action**: In each decision step i , the agent selects one frame data to transmit. The action is given by $a_i^p = (n_i, k_i) \in \mathcal{A}_i^p = \mathcal{K}_i$. Given an action, the number of RBs allocated to the selected frame is $\min(\lceil \frac{\bar{r}_{n,k,i}}{c_{n,i}} \rceil, N_i^{rb})$.

3) **Transition Probability**: Given the current state s_i and action a_i , the transition probability $P(s_{i+1}^p | s_i^p, a_i^p)$ can be defined. As depicted in Fig. 3, there are three types of state transitions.

- **Type 1: Successful Transmission**. With the chosen action, one frame is successfully transmitted regardless of whether RBs are used up.
- **Type 2: Resource Exhaustion**. With the chosen action, RBs are used up, while the selected frame is not successfully transmitted.
- **Type 3: Slot Transition**. When RBs are used up or the frame set is empty, the state transitions to a new state of the new slot by updating $w_{n,k,i}$, $\tau_{n,k,i}$, $c_{n,t}$, N_i^{rb} , and the frame set K_i (drop frames with RFDB $\tau_{n,k,i} = 1$ and add arrived frames).

For type 1 and type 2, the transition probability $P(s_{i+1}^p | s_i^p, a_i^p)$ is 1. For type 3, the transition probability $P(s_{i+1}^p | s_i^p, a_i^p)$ depends on the frame arrival distribution, and the action does not affect the state transition.

4) **Reward**: Under the selected action $a_i^p = (n_i, k_i)$ at the decision step i , the reward function can be obtained based on the types of state transitions. Let $\tilde{\mathcal{K}}_{n,t,i}$ denote the set of dropped frames in decision step i . Therefore, we formulate the reward function as,

$$r^p(s_i^p, a_i^p) = \begin{cases} 0, & \text{If type 1 or type 2,} \\ -\sum_{n \in \mathcal{N}} \sum_{k \in \tilde{\mathcal{K}}_{n,t,i}} w_{n,k,i}, & \text{If type 3.} \end{cases} \quad (14)$$

In this problem, the agent selects the highest priority frame for transmission. This policy determines the probability of the agent choosing each action at a given state, and can be

expressed as $\pi^p(a_i^p, s_i^p)$. The criterion for selecting an action based on priority is similar to the policy used in MDP.

Given a policy π , the on-policy value function is defined as the expected total discounted reward,

$$V_\pi(s^p) = \mathbb{E}_{a_i^p \sim \pi(\cdot | s_i^p)} \left[\sum_{i=0}^{+\infty} (\gamma^p)^i r^p(s_i^p, a_i^p) | s_0^p = s^p \right]. \quad (15)$$

The corresponding on-policy action-value function can be given by,

$$\begin{aligned} Q_\pi^p(s^p, a^p) &= \mathbb{E}_{s_{i+1}^p \sim P(\cdot | s_i^p, a_i^p)} \left[\sum_{i=0}^{+\infty} \gamma^i r^p(s_i^p, a_i^p) | s_0^p = s^p, a_0^p = a^p \right] \\ &= \mathbb{E}_{s_{i+1}^p \sim P(\cdot | s_i^p, a_i^p)} [r(s_i^p, a_i^p) \\ &\quad + \gamma \mathbb{E}_{a_{i+1}^p \sim \pi(\cdot | s_{i+1}^p)} [Q_\pi(s_{i+1}^p, a_{i+1}^p)]] \end{aligned} \quad (16)$$

According to (16), when the policy π is different, the action-value function is also different. We define the optimal action-value function as $Q_\pi^{p*}(s^p, a^p) = \max_{\pi} Q_\pi^p(s^p, a^p)$. Based on $Q_\pi^{p*}(s^p, a^p)$, the optimal policy can be given by,

$$\pi^{p*}(s_i^p, a_i^p) = \begin{cases} 1, & \text{if } Q_\pi^p(s_i^p, a_i^p) = \max_{a' \in \mathcal{A}_i^p} Q_\pi^{p*}(s_i^p, a'), \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

Therefore, the Bellman equation for the optimal action-value function is,

$$\begin{aligned} Q_\pi^{p*}(s_i^p, a_i^p) &= \mathbb{E}_{s_{i+1}^p \sim P(\cdot | s_i^p, a_i^p)} [r(s_i^p, a_i^p) \\ &\quad + \gamma \max_{a' \in \mathcal{A}_{i+1}^p} Q_\pi^{p*}(s_{i+1}^p, a')]. \end{aligned}$$

However, the optimal action-value function can not be obtained directly, because of the lack of prior information on transition probabilities $P(s_{i+1}^p | s_i^p, a_i^p)$. A Q-learning method based on off-policy temporal difference is proposed to estimate the optimal action-value function $Q^*(s_i, a_i)$ iteratively by,

$$\begin{aligned} Q^p(s_i^p, a_i^p) &= Q^p(s_i^p, a_i^p) + \alpha [r^p(s_i^p, a_i^p) \\ &\quad + \gamma^p \max_{a' \in \mathcal{A}_{i+1}^p} Q(s_{i+1}^p, a') - Q^p(s_i^p, a_i^p)]. \end{aligned} \quad (18)$$

The neural network of DQN is parameterized by weights and biases denoted as θ^p . To estimate the neural network, we can optimize the following loss functions $L^p(\theta_j^p)$ at iteration j ,

$$L^p(\theta_j^p) = \mathbb{E} \left[\left(y(\hat{\theta}_j^p) - Q^p(s_i^p, a_i^p; \theta_j^p) \right)^2 \right], \quad (19)$$

where

$$y(\hat{\theta}^p) = r^p(s_i^p, a_i^p) + \max_{a' \in \mathcal{A}_{i+1}^p} Q^p(s_{i+1}^p, a'; \hat{\theta}_j^p), \quad (20)$$

and $\hat{\theta}_j^p$ is the parameter of a fixed target network at each iteration j [33]. Therefore, the specific gradient update can be given by,

$$\nabla_{\theta_j^p} L(\theta_j^p) = \mathbb{E} \left[\left(y(\hat{\theta}_j^p) - Q(s_i^p, a_i^p; \theta_j^p) \right) \nabla_{\theta_j^p} Q(s_i^p, a_i^p; \theta_j^p) \right].$$

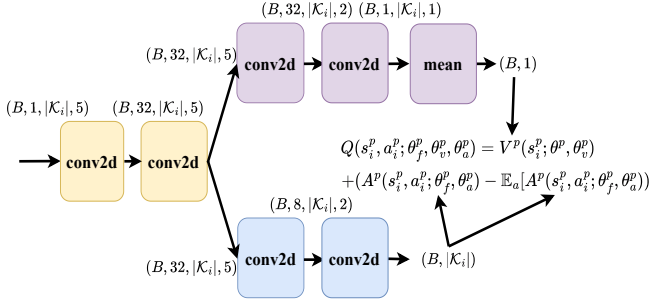


Fig. 4. 2D-CNN-based neural network design. We estimate the value function and the advantage function respectively, then obtain the action value function.

C. DQN Model Design and Training

In our algorithm, two improvements are made to the original algorithm to develop an algorithm that is tailored to this task. Firstly, as previously mentioned, there are three types of state transitions $\pi^p(s_i^p, s_i^p)$. In types 1 and 2, rewards depend on both states and actions, while in type 3, they depend only on states. In order to better estimate the rewards brought by states and actions, we decouple the action-value function into a value function and an advantage function that quantifies the advantage of each action. More specifically, we have,

$$Q^p(s_i^p, a_i^p) = V^p(s_i^p) + A^p(s_i^p, a_i^p), \quad (21)$$

where $A^p(s_i^p, a_i^p)$ denotes the advantage function. To implement this decoupling, we use the Dueling DQN method [33], and design the last module of the network as follows,

$$Q^p(s_i^p, a_i^p; \theta_f^p, \theta_v^p, \theta_a^p) = V^p(s_i^p; \theta_v^p) + (A^p(s_i^p, a_i^p; \theta_f^p, \theta_a^p) - \mathbb{E}_a[A^p(s_i^p, a_i^p; \theta_f^p, \theta_a^p)]),$$

where θ_f^p , θ_v^p and θ_a^p are parameters of feature extraction network, value network, and action network, respectively.

Another challenge is that the input and output sizes of the network are not fixed. As mentioned earlier, the input size of the network is $(B, 1, |K_i|, 5)$, and the output size is $(B, |K_i|)$, where B is the batch size. However, $|K_i|$ can have different values in different states. The size of the action space and state space is dynamically changing. To overcome this, one solution is to use an upper bound K for $|K_i|$ (where $K \geq K_i$) and pad the remaining input parameters with zeros [34], or train different models for different input sizes directly [35]. These model design methods are not always general enough to be deployed in real systems.

Combining these two aspects, we designed our model based on 2D Convolutional Neural Network (2D-CNN) layers, as shown in Fig. 4. Two 2D-CNN layers extract features of the input state and output intermediate features with size $(B, 32, |K_i|, 5)$. The value function network θ_v^p consists of two 2D-CNN layers and one mean-pooling layer, and the advantage function network θ_a^p consists of two 2D-CNN layers.

In each decision step, we store the agent's experience $(s_i, a_i, s_{i+1}, r(a_i, s_i))$ in the replay buffer. When training the neural network, we sample data in batches from the replay buffer. We train the network using the ϵ -greedy policy [33].

Specifically, when the state transitions to type 3 and $K_i = \emptyset$, we set the action to a default action $a_i = (0, 0)$, and let $Q(s_i, a_i) = 0$.

VI. LEARNING-BASED XR VIDEO ADAPTATION IN APPLICATION LAYER

This section presents a learning-based adaptive bitrate algorithm for XR video applications. The adaptive bitrate algorithm performs bitrate selection at a large time scale in order to suit dynamic channel conditions. However, as a continuous action problem, we propose using a Transformer-based PPO algorithm to adjust video bitrate.

A. Adaptive Bitrate Problem Formulation

According to the *Problem 1*, with a given resource allocation policy $\pi^p(\cdot)$, we can reformulate the problem as an adaptive bitrate problem for solving $\pi^b(\cdot)$,

Problem 3 (Adaptive Bitrate Subproblem): The adaptive bitrate subproblem can be represented as,

$$\begin{aligned} & \underset{\pi^b(\cdot)}{\text{maximize}} && \sum_{n \in \mathcal{N}} \mathbb{E}[\log \mathcal{Q}_{n,t_b}^p + \mathcal{Q}_{n,t_b}^t], \\ & \text{s.t.} && (7), (8), (9), (12) \end{aligned} \quad (22)$$

The adaptive bitrate problem aims to maximize the sum of the expected logarithm of the video quality and the transmission quality. We also consider the fairness of bitrate allocation to multiple users.

To solve this problem, we propose a learning-based adaptive bitrate selection algorithm using the TPPO algorithm. The Transformer is used to extract historical semantic features related to the environment, enabling TPPO to choose the appropriate video bitrate utilizing the historical information. The TPPO algorithm is a type of stochastic policy gradient algorithm, which employs two modules, i.e., the actor network and the critic network. The actor network maximizes the objective function and uses the gradient descent algorithm to update the policy model. The critic network evaluates the objective function based on reward parameters, using the evaluation result as the reference for further performance improvement. By iteratively updating these networks, our algorithm can learn a good policy for bitrate selection in XR video applications.

B. POMDP formulation of Problem 3

Considering that the XR server cannot obtain all state information of the BS, We formulate the *Problem 3* as a POMDP. The more detailed POMDP definition of the adaptive bitrate problem is as follows,

1) **Observed State:** In XR server, the state $s_{t_b}^b \in \mathcal{S}^b$ is partially observable. Therefore, we define the observed state at bitrate adjustment period t_b as $\mathbf{o}_{t_b}^b = (\hat{N}_{t_b}^{rb}, U_{n,t_b}^{rb}, C_{n,t_b}, |\mathcal{K}_{n,t_b}|, |\tilde{\mathcal{K}}_{n,t_b}|, W_{n,t_b}, \tilde{W}_{n,t_b}, n \in \mathcal{N}) \in \mathcal{O}^b$. The observed state $\mathbf{o}_{t_b}^b$ includes the average bandwidth $\hat{N}_{t_b}^{rb}$, RB utilization U_{n,t_b}^{rb} , average rate C_{n,t_b} , the number of transmitted frames $|\mathcal{K}_{n,t_b}|$, the number of failed frames $|\tilde{\mathcal{K}}_{n,t_b}|$, the importance weights sum of the transmitted frames W_{n,t_b}

and the importance weights sum of the failed frames $\widetilde{W}_{n,t_b}$. These are calculated as follows,

$$\hat{N}_{t_b}^{rb} = \sum_{t=t_b T_b}^{t_b T_b + T_b - 1} \frac{N_t^{rb}}{T_b}, \quad (23)$$

$$U_{n,t_b}^{rb} = \sum_{t=t_b T_b}^{t_b T_b + T_b - 1} \sum_{(n,k) \in \mathcal{K}_{n,t}} N_{n,k,t}^{rb}, \quad (24)$$

$$C_{n,t_b} = \sum_{t=t_b T_b}^{t_b T_b + T_b - 1} \sum_{(n,k) \in \mathcal{K}_{n,t}} c_{n,t} N_{n,k,t}^{rb}, \quad (25)$$

$$W_{n,t_b} = \sum_{(n,k) \in \mathcal{K}_{n,t_b}} w_{n,k}, \quad (26)$$

$$\widetilde{W}_{n,t_b} = \sum_{(n,k) \in \widetilde{\mathcal{K}}_{n,t_b}} w_{n,k}. \quad (27)$$

2) **Action**: Action is the adjustment policy of the video bitrate. At bitrate adjustment period t_b , the action is denoted by $\mathbf{a}_{t_b}^b = (\Delta R_{1,t_b} \cdots \Delta R_{N,t_b})$, where $\Delta R_{n,t_b} \in [-\Delta R^{max}, \Delta R^{max}]$ is the adjustment value with limited range. Then the bitrate after adjustment is $R_{n,t_b} = R_{n,t_b-1} + \Delta R_{1,t_b}$.

3) **Transition Probability**: With state $\mathbf{s}_{t_b}^b$ and action $\mathbf{a}_{t_b}^b$, the transition probability can be given by $P(\mathbf{s}_{t_b+1}^b | \mathbf{s}_{t_b}^b, \mathbf{a}_{t_b}^b)$. POMDP also has partial observations $\mathbf{o}_{t_b}^b$, which are conditioned on the conditional observation probability $P(\mathbf{o}_{t_b}^b | \mathbf{s}_{t_b}^b)$. Therefore, the transition distribution under the policy π_{θ^b} can be expressed as,

$$P(\Gamma | \pi^b) = P(\mathbf{s}_0^b) P(\mathbf{o}_0^b | \mathbf{s}_0^b) \prod_{t_b \in \mathcal{T}_b \setminus \{0\}} P(\mathbf{o}_{t_b}^b | \mathbf{s}_{t_b}^b) \prod_{t_b \in \mathcal{T}_b \setminus \{0\}} P(\mathbf{s}_{t_b}^b | \mathbf{s}_{t_b-1}^b, \mathbf{a}_{t_b-1}^b) \prod_{t_b \in \mathcal{T}_b \setminus \{0\}} \pi_{\theta^b}(\mathbf{a}_{t_b-1}^b | \mathbf{h}_{t_b-1}^b), \quad (28)$$

where $\Gamma = (\mathbf{s}_0^b, \mathbf{o}_0^b, \mathbf{a}_0^b, \cdots, \mathbf{s}_{|T_b|-1}^b, \mathbf{o}_{|T_b|-1}^b)$ is the trajectory, and $\mathbf{h}_{t_b}^b = (\mathbf{o}_0^b, \mathbf{a}_0^b, \cdots, \mathbf{o}_{t_b-1}^b, \mathbf{a}_{t_b-1}^b, \mathbf{o}_{t_b}^b)$ is the historical sequence.

4) **Reward**: Before the bitrate agent makes the $(t_b + 1)^{th}$ decision, the XR bitrate controller can use the information from the XR server and the RAN monitor to calculate the reward. According to *Problem 3*, the reward function can be expressed as,

$$r_{t_b}^b = \sum_{n \in \mathcal{N}} (\log \mathcal{Q}_{n,t_b}^p + \mathcal{Q}_{n,t_b}^t). \quad (29)$$

$\mathcal{Q}_{n,t_b}^p$ can be obtained from the local state, and $\mathcal{Q}_{n,t_b}^t$ is calculated from the BS information collected in real-time.

The theory of policy gradient provides a way to optimize the parameters of a policy to maximize the expected return. In POMDP, the expected return is defined as the sum of rewards obtained by following a policy over a sequence of time steps. The update direction for the policy parameters is given by the gradient of the objective function. We derive the theory of policy gradient theory and ensure that the update direction is correct for the POMDP. The overall expected return is defined

as,

$$J^b(\pi_{\theta^b}) = \mathbb{E}_{\Gamma \sim \pi_{\theta^b}} [R(\Gamma)] = \sum_{\Gamma} P(\Gamma | \pi_{\theta^b}) R(\Gamma). \quad (30)$$

where $R(\Gamma) = \sum_{t_b=0}^{\infty} \gamma^{t_b} r_{t_b}^b$, γ is the discount factor. In order to obtain a policy π_{θ^b} that maximizes the expected return $J^b(\pi_{\theta^b})$, we update the parameters θ^b using the gradient of the objective function. The correct update direction is given by,

Theorem 1: Given a function approximator Q_{η}^b , if the following conditions are satisfied,

- 1) $Q_{\eta}^b(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b) = \nabla_{\theta^b} \log \pi_{\theta^b}(\mathbf{a}_{t_b}^b | \mathbf{h}_{t_b}^b)^T \mathbf{w}$,
 - 2) the parameters $\mathbf{w}$ are obtained by minimizing the mean-squared error, $\mathbb{E}_{\Gamma \sim \pi_{\theta^b}} [(Q_{\eta}^b(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b) - Q_{\theta^b}^b(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b))^2]$,
- the policy gradient of POMDP is given by,

$$\nabla_{\theta^b} J^b(\pi_{\theta^b}) = \mathbb{E}_{\Gamma \sim \pi_{\theta^b}} [\nabla_{\theta^b} \log \pi_{\theta^b}(\mathbf{a}_{t_b}^b | \mathbf{h}_{t_b}^b) Q_{\eta}^b(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b)].$$

Proof: Please refer to Appendix A.

Therefore, the actor-critic architecture can also be extended to the partial observations domain.

C. Model Design and Training

TPPO is a variant of the PPO [36] algorithm. For both the actor network and the critic network, we use a 6-layer Transformer encoder to build feature extraction and then use a fully connected layer for outputting results. The input size of the Transformer encoder is $7N$, and the head of self-attention is set to the number users N . The size of output layer of the actor network and the critic network are N and 1, respectively. We sequentially input the information in the observation window into the actor network and critic network to obtain features at different time scales.

Similar to PPO [36], TPPO uses a clipped objective function to update the policy parameters, which ensures stable policy improvement while preventing large policy updates. TPPO utilizes the Transformer model to extract semantic information from the historical state, including network status and bitrate control information. By leveraging this information, we can perform video bitrate control and estimate the state-value function, which is critical for achieving optimal performance in adaptive real-time XR bitrate streaming. With the use of the Transformer model, our TPPO algorithm is able to capture complex temporal dependencies in the input data and achieve superior performance compared to existing bitrate adaptation algorithms.

The TPPO algorithm also uses the importance sampling method [36], and obtains the training data through old actor ($\hat{\theta}^b$) and critic network ($\hat{\eta}$). Therefore, the loss of the actor network is,

$$L^b(\theta^b) = -\mathbb{E}[\min(\frac{\pi_{\theta^b}^b}{\pi_{\hat{\theta}^b}^b} A_{t_b}(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b), \text{clip}(\frac{\pi_{\theta^b}^b}{\pi_{\hat{\theta}^b}^b}, 1 - \epsilon, 1 + \epsilon) A_{t_b}(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b))] \quad (31)$$

where ϵ is a hyperparameter, and $\text{clip}(\cdot)$ is the clip function. The clip function makes the two distributions not much

different, and guarantees the update rate is under control. $A_{t^b}(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b)$ is the generalized advantage function, and can be expressed as,

$$A_{t^b}(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b) = Q^b(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b) - V_{\eta}^b(\mathbf{h}_{t_b}^b) \quad (32)$$

$$\text{where } Q^b(\mathbf{h}_{t_b}^b, \mathbf{a}_{t_b}^b) = \sum_{t'_b=t_b}^{+\infty} (\gamma^b \phi)^{t'_b-t_b} \delta_{t'_b}^b + V_{\eta}^b(\mathbf{h}_{t_b}^b),$$

$$\delta_{t_b}^b = r_{t_b}^b + \gamma_b V_{\eta}^b(\mathbf{h}_{t_b+1}^b) - V_{\eta}^b(\mathbf{h}_{t_b}^b),$$

ϕ is used to adjust the bias-variance tradeoff, and $V_{\eta}^b(\mathbf{h}_{t_b}^b)$ is the state-value function estimated by the critic network. Therefore, we train the critic network by minimizing the mean squared error of the state value function,

$$L^b(\hat{\eta}) = \mathbb{E} \left[\sum_{\tau=0}^{+\infty} \gamma_b^{\tau} r_{t_b+\tau}^b - V_{\hat{\eta}}^b(\mathbf{h}_{t_b}^b) \right]. \quad (33)$$

VII. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed algorithm. For all results, unless specified otherwise, the bandwidth of each RB is 360KHz, the transmission power of each RB is 10 mW. We consider the pathloss of the Urban Micro cell deployment scenario [37], and the noise power spectral density is -174 dBm/Hz. Users are distributed in the area within [500m, 500m] randomly. For the XR video, the bitrate of the XR video is assumed to be between 1Mbps and 30Mbps, and the frame rate is 60FPS. Therefore, the mean size of each frame is 250Kbits. The jitter of frame size and arrival time is generated according to [6]. I-frames are transmitted periodically every $K = 4$ frame, while the other frames are P-frames. The average size ratio between one I-frame and one P-frame is $\alpha = 1.5$ [6] and the importance weights of I-frames and P-frames are set to 1 and 0.1, respectively. The FDB is set to be 10ms. The distribution of the first frame's arrival time can be represented as $t_{n,0} \in [0, 1000/f_n]$ ms. The length of one slot Δt is 1 ms, and the time interval of bitrate adjustment period is 100 ms. For the QoE model, we set $\alpha_p = 0.2$, $\alpha_t = 0.8$, $N_t = 150$, $\beta_1 = 60$, $\beta_2 = \frac{1}{2}$. For the proposed MS-DQN algorithm, the discount factor $\gamma^p = 0.9$, and the learning rate is 0.001. For the proposed TPPO algorithm, the discount factor $\gamma^b = 0.9$, $\epsilon = 0.2$, $\phi = 1$, and the learning rate of the actor network and the critic network is 0.0002 and 0.0001, respectively.

A. Performance Comparison for BS Scheduling

To compare BS scheduling algorithms, we fix the bitrate of XR video to 20Mbps, and compare the proposed MS-DQN algorithm with baseline scheduling algorithms. Results are obtained through a 2000-slot simulation. We consider the following baseline algorithms.

1) **DDPG** [8]: The DDPG algorithm scheduler uses a neural network to adjust the off-policy deterministic policy gradient algorithm based on an Actor-Critic framework.

2) **PF** [9]: The PF scheduler considers the user's current rate and historical average rate to calculate the priority.

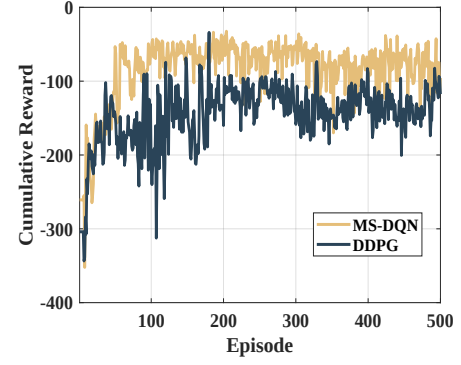


Fig. 5. Convergence performance of MS-DQN and DDPG algorithms for wireless resource allocation problem.

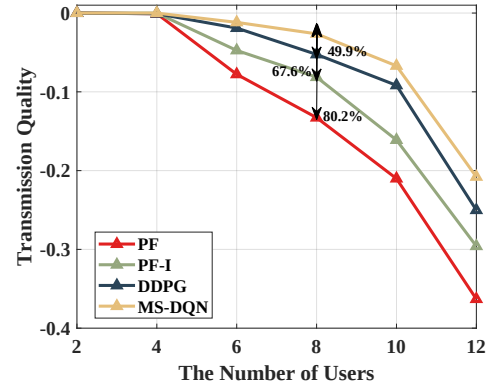


Fig. 6. Performance comparison of transmission quality between the proposed MS-DQN algorithm and baseline algorithms.

3) **PF-I** [10]: The Frame-level integrated based PF scheduler is designed to take into account both the integrity of the frame and the network channel conditions for users.

We first compare the convergence performance between our proposed MS-DQN algorithm and the DDPG algorithm. However, when the number of users and the initial time of the first frame are different, the convergence curve will also be different. For the fairness of the comparison, in each episode, we train the neural network using scenarios with varying numbers of users and initial times and test with the same scenario to observe changes of cumulative reward. Each episode of training and testing is 2000 slots. Fig. 5 depicts the cumulative reward of different algorithms. The number of users is 8. Both the proposed MS-DQN-based algorithm and the DDPG algorithm can converge after about 50 episodes. However, compared with the DDPG algorithm, the MS-DQN algorithm's cumulative reward is higher, which shows the proposed MS-DQN algorithm performs better than DDPG.

Fig. 6 shows the comparison of transmission quality among different algorithms under different numbers of users. We repeat the experiment 50 times, and each experiment is performed for 2000 time slots. As shown in Fig. 6, the proposed MS-DQN algorithm outperforms the baseline algorithm. For example, when the number of users is 8, compared with PF, PF-I, and DDPG algorithms, the proposed MS-DQN has an improvement of 80.2%, 67.6%, and 49.9% in transmission

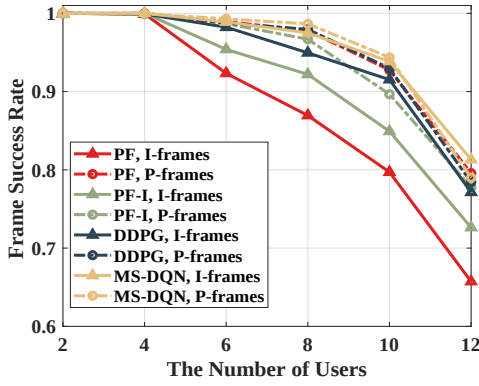


Fig. 7. I-frames and P-frames transmission performance comparison between the proposed MS-DQN algorithm and baseline algorithms.

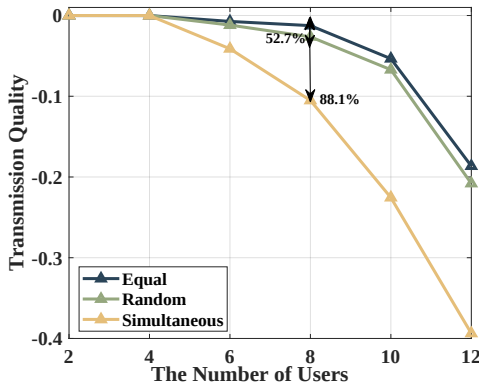


Fig. 8. Performance comparison of transmission quality with the different initial arrival time. It indicates that the distribution of frame arrivals will affect the transmission quality.

quality, respectively.

Fig. 7 illustrates the impact of different algorithms on the frame success rates of I-frames and P-frames. Specifically, for I-frame transmission, compared to the PF, PF-I, and DDPG algorithms, the proposed MS-DQN algorithm can improve the frame success rate by 10.6%, 5.3%, and 2.5% respectively when the number of users is 8. For P-frame transmission, MS-DQN achieves a frame success rate improvement of 0.7-2.0% when the number of users is 8.

Considering the impact of the initial arrival time, we test three different scenarios and plot Fig. 8. We consider three scenarios. Scenario 1: the initial frame arrival follows a uniform distribution. (Random); Scenario 2: the initial frames of all users arrive simultaneously (Simultaneous); Scenario 3: the initial frames of all users arrive with equal intervals (Equal). As shown in Fig. 8, when the number of users is 8, compared with the Random and Simultaneous settings, the Equal setting can achieve an 88.1% and 52.7% increase in transmission quality, respectively. This indicates that the frame arrival interval of different users can affect the frame success rate of BS transmission. XR video servers can use this feature to arrange and adjust encoders to optimize real-time XR video transmission.

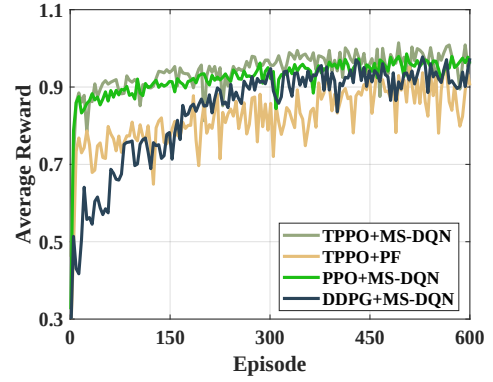


Fig. 9. Convergence performance of TPPO+MS-DQN and baseline algorithms.

B. Performance Comparison for Bitrate Adaption

Then we compare the TPPO algorithm with the baseline algorithm and obtain the QoE optimization results. The experimental results were obtained through simulation over 1000 bitrate adjustment periods, with each bitrate adjustment period consisting of 100 slots. We consider following three baseline algorithms,

- 1) **PPO [12]**: The PPO algorithm uses a fully connected network instead of the Transformer architecture.
- 2) **DDPG [11]**: Unlike the PPO algorithm, DDPG uses deterministic policy gradient to obtain the optimal actions.
- 3) **Search**: The Search algorithm assumes that the XR bitrate controller has all prior information before the next bitrate adjustment period, including the maximum number of RBs and the channel state information. It uses the binary search algorithm to select the bitrate that maximizes the current QoE. It is important to note that the Search algorithm can only provide an approximation of optimal performance as a reference. The unavailability of prior information in practical systems makes it unusable.

Fig. 9 shows the convergence of algorithms with Episodes. We train the networks of TPPO using randomly generated channels and XR video parameters, then test it using fixed channels and XR video parameters to ensure fairness in comparison. We execute a test after every 4 training episodes. Average QoE represents the mean QoE for each user during each bitrate adjustment period, with TPPO+MS-DQN delivering the highest QoE performance. Fig. 9 shows that TPPO+MS-DQN, PPO+MS-DQN and TPPO+PF algorithms converge after around 200 episodes, while DDPG+MS-DQN reaches convergence after around 300 episodes. This implies that TPPO is capable of achieving better convergence. Although the convergence speed of PPO is similar to that of TPPO, the performance of the PPO algorithm is slightly worse than that of TPPO.

Fig. 10 shows the comparison of the average QoE under different numbers of users. As shown in the figure, the proposed TPPO+MS-DQN algorithm performs the best compared to the baseline algorithm. When the number of users is less than or equal to 4, the average QoE of users is similar because the BS can satisfy real-time XR video transmission with the maximum bitrate. With the increase in user number,

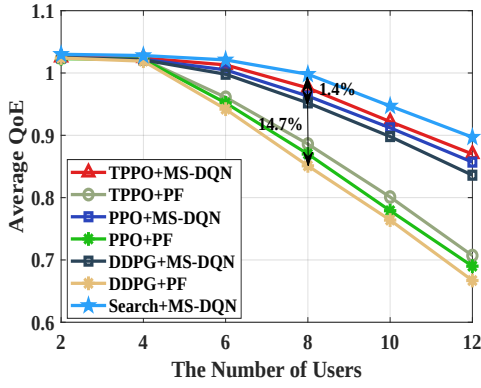


Fig. 10. The comparison of average QoE results under the different number of users.

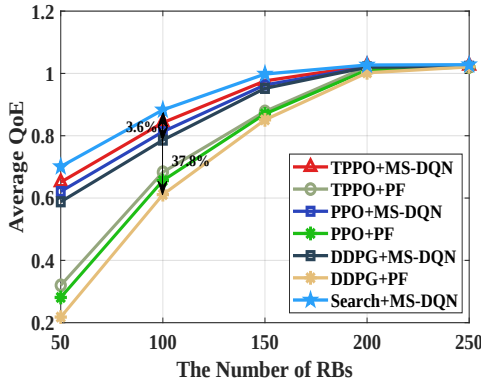


Fig. 11. The comparison of average QoE results under the different number of RBs.

competition for resources ultimately results in a decrease in the mean QoE. When the number of users is 8, the proposed TPPO+MS-DQN algorithm improves the average QoE by 1.4%-14.7% compared to the baseline algorithms. Specifically, the TPPO+MS-DQN algorithm enhances the average QoE by 14.7% compared to the DDPG+PF algorithm. Compared with the PPO+MS-DQN algorithm, the proposed algorithm improved the average QoE by 1.4% using the transformer architecture. The search algorithm outperforms the TPPO algorithm as it can choose the optimal video bitrate based on prior knowledge of the network environment. The search algorithm can approach the optimal solution, but it cannot be used in practical systems because the practical systems are unable to obtain network information about the present and future. When the number of users is 8, the TPPO algorithm can achieve a 97.8% average QoE of the Search algorithm.

Fig. 11 shows the variation of average QoE under the different number of RBs (different bandwidths). The number of users is 8. As the figure shows, a larger bandwidth brings gains in the average QoE. When the number of RBs is greater than 200, the QoE of users remains constant, as the bandwidth can meet the transmission requirements of users' maximum bitrate. When the number of RBs is 100, the proposed TPPO+MS-DQN algorithm can achieve a performance improvement of 3.6%-37.8% and an 95.4% average QoE of the Search algorithm.

In Fig. 12, we plot the average QoE and the components

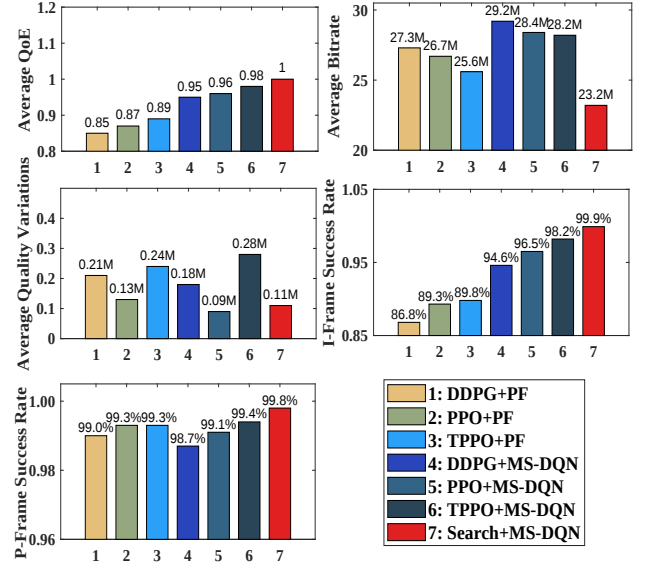


Fig. 12. Comparison of the various components of QoE when the number of users is 8 and the number of RBs is 150.

that makeup QoE, including average bitrate, average quality variation, and frame transmission success rate. The proposed TPPO+MS-DQN algorithm can achieve an average bitrate of 28.2 Mbps, an average quality variation of 0.28 Mbps, and the success rates of transmitting I-frames and P-frames are 98.2% and 99.4%, respectively. The I-frame success rate of the DDPG+PF, PPO+PF and TPPO+PF are 86.8%, 89.3% and 89.8%, which is lower than those for algorithms based on MS-DQN, because the PF algorithm doesn't consider the frame-priority. Compared with the DDPG+MS-DQN algorithm, the proposed TPPO+MS-DQN algorithm has a decrease in the average bitrate of 1.0 Mbps, but the success rates of transmitting I-frames and P-frames are increased by 3.6% and 0.7%, respectively. It is acceptable because frame transmission failures often result in greater user experience loss. Compared with the PPO+MS-DQN algorithm, the proposed TPPO+MS-DQN algorithm increases the transmission success rate of I-frames and P-frames by 1.7% and 0.3%, respectively, under the condition of almost constant average bitrate. It indicates the effectiveness of the transformer architecture.

Considering the impact of the initial arrival time, Fig. 13 shows the average QoE comparison results under the different initial arrival time. We conduct tests in three scenarios, i.e., Equal, Random, and Simultaneous. As shown in Fig. 13, the arrival time of the initial frame will affect the QoE of XR video transmission. When the number of users is 8, compared with the Random and Simultaneous settings, the Equal setting can improve the QoE by 1.9% and 30.7%, respectively. Therefore, XR servers can optimize QoE among multiple users by adjusting the initial arrival time. It is equivalent to adjusting the average transmission interval between video frames of different users. It provides a new approach for optimizing the QoE in multi-user XR transmission. We will expand the related work in the future.

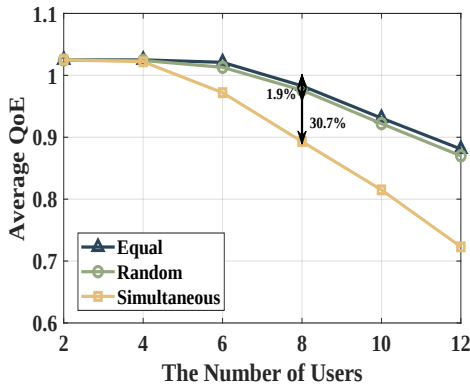


Fig. 13. The average QoE comparison results under the different initial arrival time.

VIII. CONCLUSION

In this paper, we consider cross-layer transmission optimization for real-time XR video between XR servers and BS. We design a cross-layer transmission framework that supports information interaction between BSs and XR servers to assist them in wireless resource scheduling and adaptive bitrate adjustment. We propose a frame-priority-based scheduling method and design an MS-DQN-based scheduling algorithm for the BS. We also propose a TPPO-based adaptive bitrate algorithm for the XR server, which uses the semantic information of the historical environment to select an appropriate video bitrate while considering user fairness. This multi-agent-based method solves the problem of mismatched time scales. The experiments show that our TPPO+MS-DQN algorithm can effectively improve QoE for XR video transmission in multi-user scenarios.

In addition, we demonstrate through experiments that the initial arrival time distribution of video frames affects the QoE of multi-user real-time XR video transmission. Therefore, we can optimize video transmission QoE by adjusting the initial frame arrival distribution. This research topic remains to be addressed in future work.

REFERENCES

- [1] H. Ning, H. Wang, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand, "A survey on metaverse: the state-of-the-art, technologies, applications, and challenges," *arXiv preprint arXiv:2111.09673*, 2021.
- [2] 3GPP TR 26.928, "Technical Specification Group Services and System Aspects; Extended Reality (XR) in 5G (Release 18)," V18.0.0, Mar. 2023.
- [3] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, 2020.
- [4] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. SIGCOMM'17*, 2017, p. 197–210.
- [5] G. Carlucci, L. De Cicco, S. Holmer, and S. Mascolo, "Congestion control for web real-time communication," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2629–2642, 2017.
- [6] 3GPP TR 38.838, "Technical Specification Group Radio Access Network; Study on XR (Extended Reality) Evaluations for NR (Release 17)," V17.0.0, Dec. 2021.
- [7] B. Liu, H. Tian, and L. Xu, "An efficient downlink packet scheduling algorithm for real time traffics in LTE systems," in *Proc. CCNC'2013*, 2013, pp. 364–369.
- [8] N. Sharma, S. Zhang, S. R. S. Venkata, F. Malandra, N. Mastrorade, and J. Chakareski, "Deep reinforcement learning for delay-sensitive LTE downlink scheduling," in *Proc. PIMRC'2020*, 2020, pp. 1–6.

- [9] J.-G. Choi and S. Bahk, "Cell-throughput analysis of the proportional fair scheduler in the single-cell environment," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 766–778, 2007.
- [10] E. Chen, S. Dou, S. Wang, Y. Cao, and S. Liao, "Frame-level integrated transmission for extended reality over 5G and beyond," in *Proc. GLOBECOM'2021*, 2021, pp. 1–6.
- [11] W. Liu, H. Zhang, H. Ding, Z. Yu, and D. Yuan, "Qoe-aware collaborative edge caching and computing for adaptive video streaming," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2023.
- [12] M. Naresh, P. Saxena, and M. Gupta, "PPO-ABR: Proximal policy optimization based deep reinforcement learning for adaptive bitrate streaming," in *Proc. IWCMC'2023*, 2023, pp. 199–204.
- [13] S. Shakkottai and A. L. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *Translations Amer. Math. Soc.*, vol. 207, pp. 185–202, 2002.
- [14] Y. Hao, F. Li, C. Zhao, and S. Yang, "Delay-oriented scheduling in 5G downlink wireless networks based on reinforcement learning with partial observations," *IEEE/ACM Trans. Netw.*, pp. 1–15, 2022.
- [15] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *Proc. SIGCOMM'15*, 2015, p. 325–338.
- [16] K. Spiteri, R. Ugaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1698–1711, 2020.
- [17] Y. Li, X. Zhang, C. Cui, S. Wang, and S. Ma, "Fleet: Improving quality of experience for low-latency live video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2023.
- [18] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 719–733, 2014.
- [19] H. Yuan, H. Lu, L. Meng, and M. Liu, "Muabr: Multi-user adaptive bitrate algorithm based multi-agent deep reinforcement learning," in *Proc. ICC'2022*, 2022, pp. 751–756.
- [20] N. Li, Y. Hu, Y. Chen, and B. Zeng, "Lyapunov optimized resource management for multiuser mobile video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1795–1805, 2019.
- [21] K. Tang, N. Kan, J. Zou, C. Li, X. Fu, M. Hong, and H. Xiong, "Multi-user adaptive video delivery over wireless networks: A physical layer resource-aware deep reinforcement learning approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 798–815, 2021.
- [22] J. Yu, H. Wen, G. Pan, S. Zhang, X. Chen, and S. Xu, "Quality of experience oriented adaptive video streaming for edge assisted cellular networks," *IEEE Wireless Commun. Lett.*, vol. 11, no. 11, pp. 2305–2309, 2022.
- [23] K. Xiao, S. Mao, and J. K. Tugnait, "Robust QoE-driven DASH over OFDMA networks," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 474–486, 2020.
- [24] C. He, S. Zhu, and B. Zeng, "NOMA-based uncoded video transmission with optimization of joint resource allocation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 5, pp. 2439–2450, 2023.
- [25] S. Wang, S. Bi, and Y.-J. A. Zhang, "Adaptive wireless video streaming: Joint transcoding and transmission resource allocation," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3208–3221, 2022.
- [26] D. Wang, L. Toni, P. C. Cosman, and L. B. Milstein, "Uplink resource management for multiuser ofdm video transmission systems: Analysis and algorithm design," *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 2060–2073, 2013.
- [27] S.-M. Tseng, G.-Y. Chen, and H.-C. Chan, "Cross-layer resource management for downlink bf-noma-ofdma video transmission systems and supervised/unsupervised learning based approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10744–10753, 2022.
- [28] F. Li, T. Wang, and P. C. Cosman, "Joint rate adaptation and resource allocation for real-time h. 265/hevc video transmission over uplink ofdma systems," *Multimedia Tools and Applications*, vol. 78, pp. 26807–26831, 2019.
- [29] H. Zhang, A. Zhou, and H. Ma, "Improving mobile interactive video QoE via two-level online cooperative learning," *IEEE Trans. Mobile Comput.*, pp. 1–1, 2022.
- [30] Y.-H. Jung, Q. Song, K.-H. Kim, P. Cosman, and L. B. Milstein, "Cross-layer resource allocation using video slice header information for wireless transmission over LTE," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 2024–2037, 2018.
- [31] W. Yin, L. Xu, W. Liu, Z. Cai, Y. Yang, and P. Wang, "Joint video packet assignment, power control and user scheduling over cognitive multi-homing heterogeneous noma networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4724–4735, 2022.

- [32] H. Schulzrinne, S. L. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF, July 2003.
- [33] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. ICML'2016*, 2016, pp. 1995–2003.
- [34] Y. Chen and H. Zhang, "Power allocation based on deep reinforcement learning in hetnets with varying user activity," in *Proc. GLOBE-COM'2020*, 2020, pp. 1–6.
- [35] J. Song, Q. Song, Y. Kang, L. Guo, and A. Jamalipour, "QoE-driven distributed resource optimization for mixed reality in dynamic TDD systems," *IEEE Trans. Commun.*, pp. 1–1, 2022.
- [36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [37] 3GPP TR 38.901, "Technical Specification Group Radio Access Network; Study on channel model for frequencies from 0.5 to 100 GHz (Release 17)," V17.0.0, Mar. 2022.