An Exploration-driven Reinforcement Learning Model for Video Streaming Scheduling in 5G-Powered Drone

Shuijie Wang\textsuperscript{1}, Yu Su\textsuperscript{*}, Qianqian Cheng\textsuperscript{1} and Yuhe Qiu\textsuperscript{1}

\textsuperscript{1}R&D Department II, China Mobile Chengdu Institute of Research and Development, Chengdu, Sichuan, 610000, China

\textsuperscript{*}Corresponding author’s e-mail:suyu@cmii.chinamobile.com

Abstract. In the process of wireless network video streaming, especially in more complex scenarios (such as video transmission of 5G-powered drones), analyzing the quality of experience (QoE) of the video streaming is a very crucial task. Thus attention should be paid to the dynamic interaction between QoE indicators including buffer starvation probability and traffic load. This paper proposes a video streaming scheduling model based on reinforcement learning. By learning the correlation between user behavior and traffic patterns, a series of resource allocation strategies that optimize QoE indicators are obtained. Since there is a certain degree of randomness in the network status at each moment in the transmission process, the model introduces exploration rewards to solve the noise problem of random environments. At the same time, this mechanism enables the model to fully explore the environment even when the reward is sparse, so as to obtain an effective scheduling strategy. Simulation experiments have proved that our model can improve the long-term QoE of video streaming in different network environments.

1. Introduction

Due to the rapid development of mobile video services, the delay and stability of wireless network video streaming have received more extensive attention. Unlike voice communication, video streaming applications do not have strict requirements for end-to-end delay, but it requires a set of key performance indicators (KPI) to measure the quality of experience (QoE) \[1\][2]. Among these indicators, buffer starvation probability and buffer time are particularly pivotal. They quantify the event that the buffer becomes empty. Buffer starvation will cause frozen images, that is, video freezes. Therefore, in real-time video streaming services, buffer starvation has a huge impact on user behavior. When the video is frequently in the buffering state, most users will consider terminating the video playback before the end of the transmission process \[3]\[4]\[5]\[6]\[7]\.

In order to alleviate the negative of buffer starvation, a huge amount of studies are aimed at finding a reasonable start-up delay configuration strategy. The difficulty of this problem is that wireless network video transmission is a random process that controls the arrival of data packets, so it is defective to just focus on video streaming of limited size and length\[8]\[9]\[10]. In addition, if the network status does not have geographic homogeneity, the fixed startup delay configuration is no longer applicable to different network environments \[11]\[12]. It means that the prefetching threshold calculated by a given traffic intensity and file size distribution has certain limitations, for the reason that this method ignores the influence of network status fluctuations \[13]\.

This paper proposes a video streaming scheduling model based on reinforcement learning to dynamically configure the start-up delay. The method is based on the existing starvation hunger
evaluation model, and the calculated results are regarded as part of the reward function in the training phase. Finally, the prefetching strategy that maximizes the long-term QoE and the value function of the strategy are learned. Given the exponential size of the state space, the value function can be approximated. The input of the neural network is the encoded state, and the output is the actual value of each possible action taken. The model starts to execute the strategy from a given initial state which can choose the action with the maximum value, or take random exploration actions. The network is trained based on the collected experimental data, and is performed on different threads independent of each other according to specific strategies.

The main contributions of this paper are:

- We model the wireless network video streaming transmission process as a reinforcement learning environment, and obtain the best packet prefetching strategy for long-term QoE through learning.
- We propose an exploration-driven reward mechanism, which not only solves the noise problem in the environment, but also alleviates the sparse rewards in the model training process to a certain extent.
- The reliability of this model is verified in the complex scenario of 5G-powered UAV video transmission. Experiments prove the effectiveness of the model in different network environments.

The content of this article is organized as follows. The second section summarizes the related work. The third section describes the exploration-driven reinforcement learning model for video streaming scheduling. Section 5 shows the experimental process and results. Section 6 summarizes this work.

2. Related Work
Research on the trade-off between buffer starvation and start-up delay is mainly divided into bandwidth change models and mathematical methods. Xu et al. modeled buffer starvation probability and starvation event generation function in [4], and dynamically analyzed starvation behavior at the file level. Given the distribution of traffic intensity and file size, this method can calculate the relationship between the starvation probability and the packets prefetching threshold [14][15][16][17]. Although this work is very novel in method, it only considers a single video stream of limited size and length. In [5], the authors stand from the perspective of network operators and solve related problems in three directions: measuring traffic patterns, modeling the possibility of buffer starvation and using calculation results for resource allocation [18][19]. This paper uses different short-term and long-term QoE to balance the overall QoE. At the same time, the authors also proposed a Bayesian inference method that allows operators to infer whether the input stream is short-view or long-view[20][21].

3. RL-based Video Streaming Scheduling Model
In this section, we propose a reinforcement learning model which can obtain the best packet prefetching strategy for long-term QoE ignoring the problem of environment noise and sparse rewards.

3.1. RL Environment Settings
The purpose of reinforcement learning (RL) algorithm is to learn how to control the agent to complete the specified task. In order to abstract the QoE optimizing problem as a RL problem, the behavior and environment of the agent need to be defined[22][23][24]. The environment describes the state of the task within a specified time, the set of actions that can be taken, and the impact of these actions.

The state in the environment is represented by a vector, which describes the network state and data packet state in the process of video streaming transmission in 5G NAS network. The elements in the vector include: packet arrival rate, packet service rate, duration of a service slot, start-up delay, traffic intensity, File size in packets, Packet arrival probability, Total number of packets, packet departure probability, Minimum file size, Start-up threshold in packets, Mean of exponential file size.

The action of the agent is to reconfigure the packet prefetching strategy in each state and reconfigure the startup delay when buffer starvation occurs. Every action will result in a change of the state. The state is not only determined by these actions, but also related to the current network state.
Therefore, the environment has a certain degree of randomness, and the model is a model-free RL approach.

The reward function in the environment assigns an actual value to each possible (state, action) pair. In the video stream scheduling problem, we define the reward function to be composed of two parts, which respectively represent the current state buffer starvation probability and the expected interval between two adjacent buffer starvations. This reward function is also a quantitative form of QoE.

\[ R = -(P_s + \gamma(g(E(T)))) \]  

\( P_s \) is the buffer starvation probability. The calculation of it is based on Ballot Theorem:

Ballot theorem: In a Ballot, candidate A gets \( N_A \) votes, candidate B gets \( N_B \) votes, where \( N_A > N_B \). Assuming that all orders are the same when counting votes, the probability that A will always lead in the number of votes during the whole counting process is \( (N_A - N_B)/(N_A + N_B) \).

After the start of a transmission service, for a given initial queue length \( x_1 \) and total size \( N \), the starvation probability is given by:

\[ P_s = \sum_{k=1}^{N-1} \frac{x_1}{2k-x_1} \frac{(2k-x_1)!}{k!} p^k (1-p)^{k-1} \]  

\( E(T) \) is the expected time interval between two starvations. We let \( g(\cdot) \) be a strictly mono-increasing and convex function of the expected start-up delay.

\[ E(T) = \frac{x_1}{x(1-p)} \]  

3.2. Exploration Reward Mechanism

In the actual QoE optimizing process, random environment and sparse rewards (or even almost no rewards) will cause the agent to be unable to effectively explore the environment[25][26][27]. Therefore, we propose an exploration mechanism, which is used as an intrinsic reward signal so that the agent can deal with this problem well. Our method models the exploratory reward as the difference between the predicted state and the actual state in the feature space. At the same time, the self-supervised reverse dynamic model is used to extract the state features from the feature space. When the environment changes, fine-tuning can be performed. The schematic diagram of the structure is shown in Fig. 1.

![Fig. 1. Structure of the exploration-driven reward mechanism (ERM).](image)

In the traditional RL form, the original reward signal is rewritten into \( R = R^i + R^e \), where \( R^i \) represents the intrinsic reward due to the exploration mechanism, and \( R^e \) represents the reward...
inherent in the environment, which is calculated by eq. (1). Then use the traditional strategy learning to learn the corresponding strategy by maximizing accumulated rewards.

The essence of this mechanism is to make full use of the information that actually affects the agent. We use a deep neural network to extract the feature from the state \( s \). Through which we can obtain \( \varphi(s) \). Then we use the feature extraction \( \varphi(s') \) of the next state to predict the action between these two states:

\[
a_{\text{prediction}} = DNN(s, s' ; \theta_f) \tag{4}
\]

By minimizing the error between the predicted \( a_{\text{prediction}} \) and the actual adopted action \( a \), back propagation is used to allow the neural network to learn the features that are truly affected by the action. Considering that the action here is discrete, we can take softmax function on the predicted action, and then set the corresponding loss function through maximum likelihood estimation, that is

\[
\min_{\theta_f} L_1(a_{\text{prediction}}, a) \tag{5}
\]

After extracting the feature \( \varphi(s) \) from the current state, we also use a neural network to predict the feature of the next state \( s' \):

\[
\varphi(s') = f(\varphi(s), a ; \theta_f) \tag{6}
\]

Since the predicted feature is a vector, the \( L_2 \) norm is used as the loss:

\[
L_F \left( \varphi(s'), \varphi(s') \right) = \frac{1}{2} \left| \left| \varphi(s') - \varphi(s') \right| \right|_2^2 \tag{7}
\]

At the same time, we use the above loss \( L_F \left( \varphi(s'), \varphi(s') \right) \) to construct the exploration reward:

\[
R_i = \eta L_F \left( \varphi(s'), \varphi(s') \right) \tag{8}
\]

The learning goal of the final model is given by:

\[
\min_{\theta_f} L_1(a_{\text{prediction}}, a) + \alpha E_g(s_t ; \theta_f) \sum R_i + (1 - \beta)L_1 + \beta L_F \tag{9}
\]

where \( \alpha > 0, 0 \leq \beta \leq 1 \) is only a measure of the scale of the corresponding item.

In the training phase, after the initial part of the prediction is accurate, in order to obtain more exploration rewards, this mechanism will actively explore more unknown states.

### 3.3. Optimal strategy learning

We use the approach of Deep Q Network (DQN) to learn the optimal strategy. DQN and Q-learning are similar to algorithms based on value iteration[28][29].In the ordinary Q-learning method, when the state and action space are discrete, Q-table can be used to store the Q value of each (state, action) pair. However, when the state and action space is high-dimensional continuous, using Q-table is very difficult. So we transform the Q-table update into a function fitting problem, and use the deep neural network to fit a function instead of Q-table to generate the Q value. So that the similar state can get the similar output action. The schematic diagram of the model is shown in Fig. 2.

Fig. 2. DQN algorithm diagram.
The algorithm has two main structures:

- The experience replay (experience pool) method is used to solve the problem of correlation and non-static distribution.
- A MainNet is introduced to obtain the real-time Q value, and the target Q value is obtained through another TargetNet.

The memory mechanism in the experience pool is used to learn from previous experiences. For the reason that Q-learning is an off-policy learning method, it can learn from the current experience as well as the past experience. Thus randomly adding previous experience during the learning process will make the deep neural network more efficient. The experience pool stores the transfer samples \((s_t, s'_t, r_t, s_{t+1})\) obtained by the interaction between the agent and the environment at each time step into the playback memory network, and randomly takes out some batches to disrupting the correlation while training.

Q-targets is actually a mechanism to disrupt correlation. Q-targets will build two networks with the same structure but different parameters. The network for predicting Q estimation, MainNet, uses the latest parameters, while the TargetNet parameters of the neural network that predicts Q reality use a long time ago. \(Q(s, a; \theta)\) represents the output of the current network MainNet. They were used to evaluate the value function of the current state action pair. \(Q(s, a; \theta^-)\) represents the output of TargetNet, which can be used to solve the target Q and update the MainNet parameters based on loss function. They can also copy the MainNet parameters to TargetNet after a certain number of iterations.

After that, the target Q value remains the same for a while, which can appropriately decrease the correlation between the current Q value and the target Q value to a certain degree, and furthermore, improves the robustness of the algorithm.

The update of the value function in this algorithm is given by:

\[
Q(s, a) \leftarrow Q(s, a) + \alpha [R + \gamma \max_{a'} Q(s', a') - Q(s, a)]
\] (10)

The loss function of DQN is as follows, \(\theta\) indicates that the network parameter is the mean square error loss.

\[
L(\theta) = E[(R + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta))^2]
\] (11)

There are two parts which have exactly the same structure but different parameters in DQN. The network MainNet predicts Q estimation value by latest parameters, while the TargetNet predicts Q reality value by parameters of a long time ago. It means when the agent in the model takes action \(a\) in the RL environment, Q can be calculated according to the above formula and the MainNet parameters can be updated. They will be copied to TargetNet after a certain number of iterations. Thereupon, a learning process is completed.

4. Numerical Examples

In the experiment, in order to verify that the model is effective and robust in complex 5G NAS network, a 5G-powered drone (DJI M210 UAV) equipped with a communication module (Hubble I) provided by China Mobile was used as the video streaming transmission equipment (as shown in Fig.3). The trace-driven simulation prove the accuracy of our method.

![Fig. 3. DJI M210 UAV equipped with a 5G communication module provided by China Mobile Chengdu Institute of Research and Development.](image-url)
4.1. Experiment and parameter settings

For a pure simulation experiment, the data and variables are generated according to a certain random distribution. In other words, it cannot simulate the real changing network environment, nor can it fit the noise problem in a random environment. This shows that as long as there is a large enough number of random events, pure simulation will gradually converge to a fixed mathematical model. However, this approach is not consistent with our expectations. Although pure simulation can verify the correctness of the model, it is not enough to make a satisfactory assessment of its accuracy. Therefore, this experiment considers the way of trace-driven simulation, and the video request is randomly selected from the 5G networked drone in the real wireless network environment. This can not only test the effectiveness and robustness of the model in real scenarios, but also control the range of parameters to a certain extent to achieve experimental purposes.

The experiment was carried out under the 5G NAS network, we use a 5G-powered drone to collect the video and then transmit it through a communication module. The parameters in the transmission process are returned by the module every two seconds, including the capacity of the wireless network, the maximum number of video streams that the base station can accept, the incoming stream category, and the video bit rate. At the same time, after obtaining the equal processor shared queue of the base station, the average departure rate can be obtained. In order to make the base station under a heavy load but not exceed the capacity area, we finally set the traffic intensity $\rho = 0.98$, while the video request intensity $\lambda = 0.009$. This can also make the attainment rate of the video request within a stable range.

4.2. Reward metrics and analysis

**Cumulative QoE.** In specific experiments, we will evaluate the overall objective quality of experience and the performance of the model in different start-up states of the video transmission process.

First, we divide the video stream into two categories to measure the model respectively. The Poisson arrival rate of the k-th video request is $\lambda_k$. The total arrival rate is $\lambda = \lambda_1 + \lambda_2$. The service time of a video stream in a state is equal to the video request size in bits divided by its throughput. Since viewing time follows a super-exponential distribution, the service time of each category is also exponentially distributed. Here, we use the state pair (m, n) to indicate that there are currently m first-class flows and n second-class flows passing through the bottleneck. As more video streams share bottlenecks, the probability of buffer starvation during the transmission process will increase.

Fig.4 illustrates the change in the objective QoE of the first-class stream and the second-class stream scheduling by our model when the video transmission duration increases from 0s to 3000s. Fig.5 shows the results of scheduling in the same experimental environment through the method in [5]. It can be found that although our scheduling model is unable to make the QoE index higher than the result obtained by the method in [5] at each moment of the transmission process, it can effectively improve the long-term cumulative reward during the process.

![Fig. 4. QoE vs transmission duration (ours).](image1)

![Fig. 5. QoE vs transmission duration (method in [5]).](image2)
When the video streaming transmission process started in different states, we also evaluated the fluctuation of the objective QoE, as shown in Fig. 6 and Fig. 7. We used the state pairs of (0, 6), (2, 6), (4, 8) to conduct experiments. In addition, Fig. 8 and Fig. 9 compare the long-term cumulative QoE of first-class and second-class flows with different maximum numbers of coexisting flows respectively. As the maximum number of streams increases, more video streams may coexist in the base station, thus causing buffer starvation to occur more often. Comparing the cases where the maximum flow number is 5, 10 and 15, it can be found that the long-term cumulative QoE has a strong correlation with the maximum flow amount. Therefore, our model can be used to design a video stream admission control strategy that can tolerate a certain degree of starvation probability.

Mean DT/VT ratio. The DT/DV ratio is an important indicator that reflects the best buffer ratio during the entire video streaming period. In Fig. 10 and Fig. 11, we plot the average DT/DV ratio when the maximum number of streams increases from 5 to 15. It can be seen from the curve that the increase of the amount of flows leads to a higher average DT/DV ratio. However, our scheduling strategy can effectively reduce the video buffering time compared to the method in [5] and transmission without scheduling.
Our reinforcement learning model provides an effective scheduling strategy for the QoE tradeoff of heterogeneous video streams. When different types of video streams have different perceptions of buffer starvation, our algorithm can empower the base station, so that it has the ability to intelligently schedule different flows, which can optimize the long-term cumulative QoE. For example, if a flow is more sensitive to buffer starvation behavior at a certain moment, the scheduling strategy can provide it with higher priority.

5. Conclusion
In this paper, we propose an exploration-driven reinforcement learning model for video streaming scheduling in 5G-powered UAV. This method can generate packets prefetching and start-up delay strategy dynamically, and has great adaptability to complex network environment. The past mathematical models are always limited by Random noise and unstable scenes in the network environment. Thus we perform an exploration mechanism which make full use of the information that actually affects the agent. Through which a better UAV video streaming transmission scheduling strategy in 5G NAS network can be obtained.

Acknowledgments
This work is supported by CMLC team of China Mobile Chengdu Institute of Research and Development.

References
[1] Singh R, Kumar P R. Optimal Decentralized Dynamic Policies for Video Streaming over Wireless Channels[J]. 2019.
[2] Hsieh P C, Hou I H. Heavy-Traffic Analysis of QoE Optimality for On-Demand Video Streams Over Fading Channels[J]. IEEE/ACM Transactions on Networking, 2018.
[3] Xiao Z, Xu Y, Feng H, et al. Modeling Streaming QoE in Wireless Networks with Large-Scale Measurement of User Behavior[C]// GLOBECOM 2015 - 2015 IEEE Global Communications Conference. IEEE, 2015.
[4] Analysis of Buffer Starvation With Application to Objective QoE Optimization of Streaming Services[J]. IEEE Transactions on Multimedia, 2014, 16(3):813-827.
[5] Xu Y, Xiao Z, Feng H, et al. Modeling Buffer Starvations of Video Streaming in Cellular Networks with Large-Scale Measurement of User Behavior[J]. IEEE Transactions on Mobile Computing, 2017:1-1.
[6] Qiu H, Psounis K, Caire G, et al. High-rate WiFi broadcasting in crowded scenarios via lightweight coordination of multiple access points[C]// Acm International Symposium on Mobile Ad Hoc Networking & Computing. ACM, 2016.
[7] Zhang Y, Psounis K. Efficient MU-MIMO via Switched-beam Antennas[C]// Acm International Symposium on Mobile Ad Hoc Networking & Computing. ACM, 2017.
[8] Bader A, Member S, Fellow, et al. Mobile Ad Hoc Networks in Bandwidth-Demanding Mission-Critical Applications: Practical Implementation Insights[J]. IEEE Access, 2017, PP(99):1-1.

[9] El-Azouzi R, Acharya K V, Poojary S, et al. Analysis of QoE for Adaptive Video Streaming over Wireless Networks with User Abandonment Behavior[C]// IEEE Wireless Communications and Networking Conference. IEEE, 2019.

[10] Zhou Y, Gu X, Wu D, et al. Statistical Study of View Preferences for Online Videos With Cross-Platform Information[J]. IEEE Transactions on Multimedia, 2018, 20(6):1512-1524.

[11] Sidi, Habib, B, et al. Multipath Streaming: Fundamental Limits and Efficient Algorithms[J]. IEEE Journal on Selected Areas in Communications, 2017.

[12] Sun W, Koo J, Byeon S, et al. BlueCoDE: Bluetooth coordination in dense environment for better coexistence[C]// IEEE International Conference on Network Protocols. IEEE, 2017.

[13] Yonglong Z, Konstantinpos P. Consistently High MIMO Rates via Switched-Beam Antennas[J]. IEEE/ACM Transactions on Networking, 2018, PP:1-14.

[14] Lajos Takács. A Generalization of the Ballot Problem and its Application in the Theory of Queues[J]. Journal of the American Statistical Association, 1962, 57(298):327-337.

[15] Rani P, Gupta A, Singh Y. Multiple Event Detection In Wireless Sensor Networks Using Compressed Sensing: Health Monitoring Perspective[J]. International Journal of Computers & Engineering, 2018, 6(1):36-41.

[16] Poojary S, El-Azouzi R, Altman E, et al. Analysis of QoE for adaptive video streaming over wireless networks[C]// 2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt). 2018.

[17] Johnson D B, Maltz D A. Dynamic Source Routing in Ad Hoc Wireless Networks[M]// Mobile Computing. 1996.

[18] Park J, Chung K. Queueing Theoretic Approach to Playout Buffer Model for HTTP Adaptive Streaming[J]. KSII Transactions on Internet and Information Systems, 2018, 12(8):3856-3872.

[19] Ezzeldin Y H, Sengupta A, Fragouli C. Wireless Network Simplification: The Performance of Routing[J]. IEEE Transactions on Information Theory, 2020, 66(9):5703-5711.

[20] Yang P, Zhang N, Zhang S, et al. Content Popularity Prediction Towards Location-Aware Mobile Edge Caching[J]. IEEE Transactions on Multimedia, 2019, 21(4):915-929.

[21] Wu M, Zhong Y, Wang G, et al. URLLC in Large-Scale Wireless Networks with Time and Frequency Diversities[C]// 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2020.

[22] Plessen M G. Automating Vehicles by Deep Reinforcement Learning using Task Separation with Hill Climbing[J]. 2017.

[23] Hu H, Shan H, Wang C, et al. Video Surveillance on Mobile Edge Networks—A Reinforcement-Learning-Based Approach[J]. IEEE Internet of Things Journal, 2020, 7(6):4746-4760.

[24] Forestier S, Oudeyer P Y. Towards hierarchical curiosity-driven exploration of sensorimotor models[C]// Joint IEEE International Conference on Development & Learning & Epigenetic Robotics. IEEE, 2015.

[25] Authors A. Large-Scale Study of Curiosity-Driven Learning[J]. 2018.

[26] Nasuto S J, Hayashi Y. Anticipation: Beyond synthetic biology and cognitive robotics[J]. Biosystems, 2016, 148:22-31.

[27] Ivo M, Cornelia H, Christoph B, et al. The Impact of Financial Reward Contingencies on Cognitive Function Profiles in Adult ADHD[J]. Plos One, 2013, 8(6): e67002-.

[28] Wang K, Liu Q, Chen L. On Optimality of Greedy Policy for a Class of Standard Reward Function of Restless Multi-armed Bandit Problem[J]. IET Signal Processing, 2012, 6(6):584-593.
[29] Kearns M J, Littman M L, Singh S P, et al. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System[J]. Journal of Artificial Intelligence Research, 2000, abs/1106.0676(1):105–133.