Learning-based Buffer Starvation Modeling for Packets Prefetching Strategies of Video Streaming Services

Yu Su¹, Shuijie Wang¹*, Qianqian Cheng¹ and Yuhe Qiu¹

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

*Corresponding author’s e-mail: wangshuijie@cmii.chinamobile.com

Abstract. Improving the quality of experience (QoE) of video streaming is a significant task in the wireless network scenario. Buffer starvation in the transmission process will cause playback freeze, and a certain number of packets must be prefetched before the service restarts. Taking into account the shortcomings of buffer in video streaming services, this paper proposes a deep learning-based starvation probability calculation model and a reinforcement learning-based packet prefetching model. The deep learning approach extracts the correlation between different timing inputs through the recurrent neural network module to return an explicit result and the precise distribution of the number of buffer starvation. The reinforcement learning approach leverages a better trade-off between start-up/rebuffering delay and buffer starvation by adjusting the packet prefetching strategy, so that the long-term objective quality of experience (QoE) of the video stream is optimized. Our framework can be applied to actual scenarios including finite video streaming and long video streaming transmission.

1. Introduction

Nowadays, mobile video services are developing rapidly while the stability of video streaming transmission has received more and more attention. Video streaming applications dominate the streaming traffic on mobile networks [1][2]. Compared with voice communication, video streaming applications have less stringent requirements for end-to-end delay, but it requires a set of key performance indicators (KPI) to evaluate the user's quality of experience (QoE) [3]. Among them, the buffer starvation probability is an important performance indicator, which is used to evaluate the event that the buffer is empty [4][5][6]. When buffer starvation occurs, the video enters the buffering state, and users will see frozen images at this time. Therefore, buffer starvation is not expected to occur in real-time video streaming services [7].

The main goal of this article is to find the distribution of the starvation amount in the packets transmission process, which is applicable to both finite size and infinite size video files. We first model the buffer changes as a Markov process, and then expand it to merge bursty packet arrivals. In this system, a fixed number of packets (also known as "prefetching threshold") are prefetched before the service starts or after a starvation event. This decision-making process can alleviate to a certain extent the buffer starvation of video transmission under complex and unstable network conditions, such as 5G networked drone video backhaul scenarios.

The main contributions of this paper are:

- We propose a model based on deep neural network to characterize the buffer starvation behavior in the process of video transmission, and get the exact result of starvation distribution.
We propose a model based on reinforcement learning to select the packets prefetching strategy, which can alleviate buffer starvation to a certain extent and maximize the long-term objective QoE.

The reliability of the model is verified by a tracking-driven simulation under the complex network transmission scenario of 5G networked drone backhaul video, and combined with practical considerations such as limited video streams and ultra-long video streams.

The content of this article is organized as follows. The second section summarizes the related work. The third section describes the prediction model based on deep learning. Section 4 describes the packets prefetching strategy model based on reinforcement learning. Section 5 shows the experimental process and results. Section 6 summarizes this work.

2. Related Work
There are some studies that are more relevant to our work, [7] proposed a Ballot theorem method to calculate the starvation probability in the M/M/1 queue of limited size. It provided a clear solution and is summarized as M/D/1 queue. The study also proposed a recursive method to calculate the distribution of starvation amount, and further extended it to the ON/OFF burst arrival process [8][9][10][11][12]. Given the start-up threshold, the author provided a fluid model to calculate the starvation probability and analyzed how the prefetching threshold influences the starvation probability [13][14][15][16]. Xu et al. [3] proposed a set of novel models to characterize the starvation behavior of coexisting video streams in the base station (BS). The decision algorithm at the BS can balance the starvation behavior between the short view and the long view. In addition, they also proposed a Bayesian inference method so that network operators can infer whether the input stream is short or long.

3. Buffer Starvation Behavior Prediction Model
In this section, we will study buffer starvation behavior with finite number of arrivals based on a deep neural network model.

3.1. System Description
We propose a packet-level deep learning model to calculate the probability of starvation and the distribution of starvation amount for a single video transmission. This method introduces the recurrent neural network module to extract the correlation between different time series [17][18]. At the same time, the model uses a multi-task learning structure to share a large part of the weights for the prediction, which reduces the scale of parameters and makes the prediction more efficient.

The input of the network is the state vector of different timings in the process of data packet transmission. The elements contained in the vector are summarized in Table 1, including Poisson arrival rate, Poisson service rate, and traffic intensity etc.

Table 1. The elements contained in the state vector.

| Notation | Description                  | Notation | Description             |
|----------|------------------------------|----------|-------------------------|
| λ        | Packet arrival rate          | T1       | Start-up delay          |
| μ        | Packet service rate          | d        | Duration of a service slot |
| ρ        | Traffic intensity            | N        | File size in packets   |
| p        | Packet arrival probability   | Np       | Total number of packets |
| q        | Packet departure probability | Nm       | Minimum file size      |
| x1       | Start-up threshold in packets | θ        | Mean of exponential file size |

The model is mainly divided into three parts: space attention mechanism module, bilateral gate recurrent unit module (BiGRU) [17], and multi-task learning module. The combination of the three structures can effectively extract the correlation between different elements in the same state vector and
the correlation between different timing state features. At the same time, it can also give greater weight to some elements that have a greater impact on the final buffer starvation probability, and make full use of the extracted correlations at different levels to obtain better results. The schematic diagram of the model structure is shown in Fig. 1.

![Model Structure Diagram](image)

**Fig. 1.** Structure of the buffer starvation behavior prediction model.

### 3.2. Loss Function

The loss function is represented by the sum of two parts

$$\text{Loss} = \alpha \cdot \text{MeanSquareError} + \beta \cdot \text{CrossEntropy}$$  \hspace{1cm} (1)

where $\alpha$, $\beta$ denote discount factors.

The first part of the formula is the starvation probability loss. The specific calculation is given by

$$\text{MeanSquareError} = \| Y_S - P_S \|_2$$  \hspace{1cm} (2)

In this part, $Y_S$ represents the output value of the model which fits the starvation probability, and $P_S$ is given by the following formula based on the famous Ballot theorem [11].

$$P_S = \sum_{k=1}^{N-1} \frac{x_1}{2k-x_1} p^{k-x_1} (1 - p)^k$$  \hspace{1cm} (3)

The detailed proof can be found in [7].

During the file transfer process, starvation events may occur multiple times. Given a finite file size $N$, the maximum number of starvations is $J = \lfloor N/x_1 \rfloor$, where $\lfloor \cdot \rfloor$ is the lower limit of real numbers. $P_S(j)_i$ represents the probability of meeting $j$ starvations. Therefore, the second part is the distribution loss of the starvation amount, using cross-entropy loss.

$$\text{CrossEntropy} = -\sum_{i=1}^{n} P_S(j)_i \cdot \log (Y_{Di})$$  \hspace{1cm} (4)

The vector $Y_D = (P_S(0), P_S(1), ..., P_S(J))$. We let $P_{\epsilon(k_1)}$, $P_{S_1(k_1)}$, $P_{U_1(k_1)}$ be the probabilities of events 'the buffer becoming empty for the first time in the entire path', 'the empty buffer after the service of packet given that the previous empty buffer happens at the departure of packet $k_1$' and 'the last empty buffer observed after the departure of packet $k_1$'. The calculation process of $P_S(j)$ is given by

$$P_S(j) = \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} \cdots \sum_{k_{j-1}=1}^{N} \sum_{k_{j}=1}^{N} P_{\epsilon(k_j)} \cdot P_{S_1(k_j,k_2)} \cdots \prod_{l=1}^{j-1} P_{S_l} P_{U_j}^T$$  \hspace{1cm} (5)

where $T$ denotes the transpose. The detailed analysis can be found in [7].
4. Packets Prefetching Strategy Model

This section proposes a packets prefetching strategy model based on reinforcement learning [19], which uses content prefetching as a way to achieve the best compromise between the user’s start-up delay and starvation behavior (starvation probability or continuous playback interval).

4.1. Reinforcement Learning Model Environment Settings

The purpose of the reinforcement learning algorithm is to learn to control the agent to make it perform a given task. To abstract the data packet prefetching strategy problem as a reinforcement learning problem, it is necessary to define the environment of the agent. The environment describes the state of the task at a given time, the set of actions taken by the agent, and the results of these actions [20].

States. The state of this environment is the same as that of the deep learning model in the previous section, and is represented by the state vector in the process of packets transmission.

Action. The action is to prefetch different numbers of packets. The agent performing an action will cause the state to change.

Reward function. In [7], the author configures a starting threshold to ensure that the starvation probability is less than a certain value. The bounds of the starvation probability are considered to be parameters obtained from experiments. In this article, we define an objective QoE cost function for users, and combine it with the starvation probability generation function as a reward function for reinforcement learning.

\[
\text{Reward} = -(G(z) + \gamma g(E(T)))
\]  

(6)

The probability of no starvation, \( P_S(0) \), is computed as \( 1 - P_S \) where \( P_S \) is obtained from eq. (1). And eq. (5). Since the starvation event takes non-negative integer values, we can write the probability generating function by:

\[
G(z) = E(z^j) = \sum_{j=0}^{\infty} P_S(j) \cdot z^j
\]  

(7)

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

\[
E(T) = \frac{x_1}{\lambda(1-\rho)}
\]  

(8)

4.2. Optimal strategy learning

Aiming at the packet prefetching strategy, this paper adopts the A3C algorithm as the optimal strategy for learning. The working principle of the algorithm is to establish two neural networks: Actor network and Critic network. The output of the last layer of the Actor network is the probability distribution of each action. The critic network only outputs one value, the value function of the current state. The purpose of training this network is to learn an Actor network that can reliably predict the long-term value of actions [21][22].

A3C is a policy gradient method. The key of the policy gradient method is to estimate the gradient of the expected total return by observing the state obtained by following the strategies. The gradient of the cumulative long-term reward with respect to the parameter \( \theta \) is given by

\[
\nabla_{\theta} E_\pi = \left[ \sum_{t=0}^{\infty} \gamma^t \cdot r_{t+1} \right] = E_{\pi} \left[ \nabla_{\theta} \log \pi(s,a) A^\pi(s,a) \right]
\]  

(9)

where \( \gamma \) denotes discount factor. Here, \( \pi \) is the scheduling strategy, and \( A \) is the action value function which is calculated by state and action.

In order to converge the training process, the expected return can be tracked. The expected reward in the packets prefetching problem is very low at first, but will eventually converge to the local optimal expected reward.

The model architecture of the Actor and Critic network is shown in 0. Each network is composed of multiple convolutional layers which adopts a position-invariant detection feature mode, and is connected to 3 fully connected hidden layers. The two networks are disjoint and do not share any layers.
Table 2. Actor and Critic network model structure.

| Hidden Layer | Number of Neurons | Filter Size | Activate Function |
|--------------|-------------------|-------------|-------------------|
| 2D CNN       | 48                | n×20        | LeakyReLU         |
| 2D CNN       | 64                | n×10        | LeakyReLU         |
| FNN          | 256               | NA          | LeakyReLU         |
| FNN          | 256               | NA          | LeakyReLU         |
| FNN          | 128               | NA          | LeakyReLU         |
| Actor output FNN | Number of actions | NA        | Softmax           |
| Critic output FNN | 1              | NA          | Linear            |

5. Numerical Examples

In the specific experiment, in order to verify that the model is still valid in a complex network environment, a 5G networked drone (DJI M210 UAV) equipped with a communication module provided by China Mobile was used as the experimental equipment (as shown in Fig. 2).

Fig. 2. DJI M210 UAV equipped with a 5G communication module (China Mobile Hubble I).

5.1. Starvation of transmission process

Starvation Behavior Prediction: This set of experiments compared the starvation probability calculated by the model with the results of event-driven simulations. Video files from different sizes have been tested up to 8000 times. We deliberately consider four combinations of parameters: $\rho = 0.9$ or $1.1$, $x_1 = 30$ or 50 packets. If not explicitly mentioned, the departure rate $\mu$ is normalized to 1. The file size in the experiment is between 200 and 6000 packets. Fig. 3 shows the probability of 0~4 starvations with the parameters $\rho = 0.9$ and $x_1 = 30$. As the file size increases, the probability of no starvation decreases. We observe that the probability of multiple starvations first increases and then decreases. Fig. 3 also shows that our analysis results are in great agreement with the simulation results. When the starting threshold is 50 packets, Fig. 4 shows similar results. Fig. 5 verifies the asymptotic probability of no hunger with the traffic intensity $\rho = 1.1$, $x_1 = 50$ and 100. Fig. 6 plots the asymptotic probability of a single starvation with the same parameters.

Fig. 3. Probability of 0~4 starvations with $x_1=30$. Fig. 4. Probability of 0~4 starvations with $x_1=50$. 
5.2. Optimizing objective quality of experience

Optimizing objective QoE through the packet prefetching strategy: We verified the optimal strategy calculation problem by means of specific experiments at the media server. In Fig. 7, we illustrate the total QoE cost (including starvation cost and start-up delay cost) with $\lambda = 16, 20, 24,$ and $\mu = 25$. The file size is set to $N = 200 \sim 6000$ packets, and the discount factor $\gamma$ is $10^{-3}$. We observe that the total QoE is neither concave nor convex with respect to the action ($x_1$). For example, when the action is fixed with $\lambda = 16$, the increase in the start-up delay cost cannot be compensated by the reduction of the starvation probability.

Fig. 7. Total QoE cost with $\mu=25$

Fig. 8. Compared with other methods

6. Conclusion

In this paper, we propose a deep learning model to perform an accurate packet-level analysis of the buffer starvation behavior during video transmission. This method uses an attention-based recurrent neural network to learn the starvation probability and the distribution of starvation amount at the same time. In addition, we also proposed a model based on reinforcement learning to select the packet prefetching strategy, which can relieve buffer starvation to a certain extent and maximize the long-term objective QoE.

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] 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.

[4] 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.

[5] Zhang Y, Psounis K. Efficient MU-MIMO via Switched-beam Antennas[C]. ACM International Symposium on Mobile Ad Hoc Networking & Computing. ACM, 2017.

[6] Y. D. Xu and E. Altman et al., “Probabilistic analysis of buffer starvation in Markovian queues,” in Proc. IEEE Infocom 2012, tech. rep.

[7] Analysis of Buffer Starvation With Application to Objective QoE Optimization of Streaming Services[J]. IEEE Transactions on Multimedia, 2014, 16(3):813-827.

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

[9] 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.

[10] Hou I H, Hsieh P C. QoE-Optimal Scheduling for On-Demand Video Streams over Unreliable Wireless Networks[C]. the 16th ACM International Symposium. ACM, 2015.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Luan T H, Cai L X, Shen X S. Impact of Network Dynamics on User’s Video Quality: Analytical Framework and QoS Provision[J]. IEEE Transactions on Multimedia, 2009, 12(1):64-78.

[16] 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.

[17] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer ence, 2014.

[18] Donahue J, Hendricks L A, Rohrbach M, et al. Long-term Recurrent Convolutional Networks for Visual Recognition and Description[C]. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017:677-691.

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

[20] Miagkikh V, & Punch W F. 1999 Giraffe: An Approach to Solving Combinatorial Optimization Problems Using a Population of Reinforcement Learning Agents GECCO ’99 Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation 529 484489

[21] 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.

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