# Reinforcement Learning Based Dynamic Adaptive Video Streaming for Multi-client over NDN

Dezheng Liu
*University of Science and Technology of China*
*Laboratory for Future Networks*
Hefei, China
ldz240730@mail.ustc.edu.cn

Xiaobin Tan
*University of Science and Technology of China*
*Laboratory for Future Networks*
*Institute of Artificial Intelligence, Hefei Comprehensive National Science Center*
Hefei, China
xbtan@ustc.edu.cn

Yangyang Liu
*University of Science and Technology of China*
*Laboratory for Future Networks*
Hefei, China
lyy2017@mail.ustc.edu.cn

Yi He
*University of Science and Technology of China*
*Laboratory for Future Networks*
*Institute of Advanced Technology, University of Science and Technology of China*
Hefei, China
heyi218@mail.ustc.edu.cn

Quan Zheng
*University of Science and Technology of China*
*Laboratory for Future Networks*
*Institute of Artificial Intelligence, Hefei Comprehensive National Science Center*
Hefei, China
qzheng@ustc.edu.cn

***Abstract*—The performance of Dynamic Adaptive Streaming (DAS) in multi-client scenarios can be improved by taking advantage of the aggregation capability of Named Data Networking (NDN). In this paper, we propose a client-side reinforcement learning based (RL) ABR algorithm for NDN that can achieve proactive aggregation of requests among clients as much as possible without requiring coordinating with other clients or scheduling by a central controller. We model the interaction process between the DAS client and the network as a Markov decision process. Then, the appropriate states and rewards are selected to decide on the Markov decision process through the reinforcement learning algorithm. Through constant training, the reinforcement learning algorithm is able to guide the client to request the same video bitrate, namely request aggregation, thereby reducing repetitive traffic and achieving fairness. Compared with the existing solutions, through experiments in multi-client video distribution scenarios, the RL algorithm performs well in the overall Quality of Experience (QoE), fairness, and aggregation rate, etc.**

***Index Terms*—Named Data Networking, Dynamic Adaptive Streaming, Multi-client, Reinforcement Learning**

## I. Introduction

In recent years, with the rapid growth of portable smart devices and online digital content, Internet traffic increases exponentially. It was reported by Cisco [1] that video traffic will account for 82% of Internet traffic by 2022 and this puts heavy pressure on the current Internet. At the same time, the users' requirements on the network for the experience when watching the video are also gradually increasing. In the face of increasing video traffic and increased network user requirements for the watching experience, the traditional point-to-point network architecture will bring about serious network congestion.

Named Data Networking (NDN)[2], as a typical one of future network architecture, is a content-centered network architecture with content addressing and routing, no address required, and there is a certain cache space in the router. Such a content-centric network architecture can reduce the number of popular server hot data, reduce the burden of the server, while alleviating network congestion, and can improve data transmission efficiency.

There is a Dyanmic Adaptive Streaming over HTTP (DASH)[3] to regulate adaptive video transmission and improve the quality of video transmission services. Adaptively choosing the most appropriate video quality level according to the current network status of the client, will maximize the use of bandwidth resources, and improve the user's experience of watching the video.

This paper investigates the problem of dynamic adaptive video streaming (DAS) transmission in the NDN. Due to the characteristics in the NDN network, when clients request the same copy of the data, only one copy of the data needs to be transmitted on the bottleneck bandwidth, which will reduce the pressure of the bottleneck bandwidth and improve the video transmission efficiency, which is the so-called request aggregation. We will make full use of the request aggregation

features in NDN to improve the user's experience of watching the video. To allow users to obtain the best video watching experience, this paper proposes dynamic adaptive algorithms based on reinforcement learning and improves interuser fairness as much as possible while considering multiple users. The main contribution of this paper is summarized as follows:

- **Reinforcement Learning Base algorithm:** Compared with the GameBase algorithm [13], our Reinforcement Learning Base algorithm has better performance, and can track the changes of the environment, while the parameters of the GameBase algorithm are fixed.
- **DAS over NDN framework based on online training:** We have designed an online training-based DAS over NDN framework for multiple clients.In this framework, through online training, the system is able to obtain various information in the environment, and then continuously optimize the decision algorithm to facilitate the optimal decision making.
- **Design and experiment of real platform:** In this paper, we implement the real experimental platform, deploy the algorithm on the real experimental platform,and compares it with other algorithms to verify the performance of the algorithm.

## II. Related works

### A. Single-client Adaptive Video Streaming

There are not so many single-client ABR algorithms specifically designed for NDN. There are some typical algorithms in HTTP that are purely client-side based [5], [8], [15], [16], [17], which can be introduced to NDN network by minor modification. Although there are lots of similarities in adaptive video streaming between NDN and HTTP based on IP networks, due to the characteristics of the NDN network architecture such as in network caching and multipath forwarding, the bitrate adaptive algorithms design for HTTP suffer from drawbacks such as large bandwidth estimation errors and low network resource utilization when applied directly to NDN.

More importantly, the performance of these single-client algorithms can not be well improved because it does not consider the capability of request aggregation in NDN. And they are prone to unfairness issues because each client makes a bitrate decision from his perspective only.

### B. Multi-client Adaptive Video Streaming

In [13], an algorithm designed for a multi-client is proposed.The algorithm is based on game theory: it modeled the multi-client scene as a non-cooperation game model, and solved the problem by designing a suitable set of utility function pairs. But their utility function pairs are designed empirically, with some limitations.

In this paper, we want to design a client-side ABR algorithm over NDN that can achieve optimal and fair QoE for multiple clients by leveraging the aggregation capability of NDN without requiring coordinating with other clients or scheduling by a central controller. And it can maintain good performance in different scenarios.

## III. System Modeling and Problem Formulation

### A. System Architecture

DAS transmission architecture in NDN is shown in Fig. 1. First the client should download the Media Presentation Description (MPD) to obtain the relevant information of the video. Then it can independently select the bitrate of the video segment and send a request to the NDN through NDN Forwarding Daemon (NFD)[4], waiting the return from the NDN network. The state collection module is able to collect state information and provide it to the training module, which trains the bitrate controller model according to the state information. And the bitrate controller selects the appropriate bitrate according to the current state. The bitrate controller contains a bitrate adaptive control algorithm that can select the adaptive bitrate according to the bandwidth information, buffer information, etc. This is also the main work of this paper.

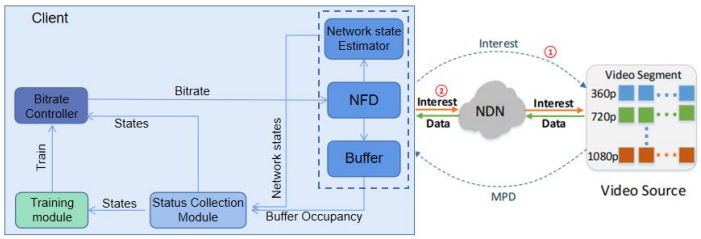


Fig. 1. Architecture of online training-based DAS over NDN.

### B. QoE Model

A widely considered metric of the performance of video streaming service is the quality perceived by the end users from enjoying the service, so-called QoE. We introduce the QoE metric proposed by [5], which is widely accepted and used in the literatures such as [6][7][8]. The QoE is calculated as follows:

$$QoE = \sum_{n=1}^{N} q_n^{\gamma} - \alpha \sum_{n=2}^{N} \frac{|q_n - q_{n-1}|}{q_n} - \beta \sum_{n=1}^{N} T_n \quad (1)$$

Among them, $N$ is the number of video clips, $q_n$ is the bitrate level of the $n$ clip, and $T_n$ is the interruption time when the n clip was downloaded. $\gamma$, $\alpha$ and $\beta$ are the weight coefficient, representing the users' attention to video quality, quality switching and interruption time. The QoE here is a cumulative metric, also known as the cumulative QoE. The bitrate decisions in the experiment are made from fragment by fragment, so this paper defines the reward after downloading one video clip as

$$r_n = q_n^{\gamma} - \alpha \frac{|q_n - q_{n-1}|}{q_n} - \beta T_n \quad (2)$$

which reflects the QoE value of each step.

### C. Problem Description

Through a comprehensive analysis of the technical architecture and user requirements of adaptive streaming in NDN, we can formulate the multi-client adaptive video streaming in NDN as a distributed optimization problem. The ultimate goal of bitrate adaptation in multi-client scenario over NDN is to maximize the overall QoE of all users watching this video, while achieving fairness among users. Since geometric mean of all users' QoE embodies both the overall efficiency and fairness of DAS system among users, our goal of bitrate adaptation can be formulated as maximizing the geometric mean of QoE of all users.

Assuming there are $N$ clients who watch the same video, the optimization problem of multi-client adaptive video streaming in NDN can be modeled as follows:

$$\begin{aligned} \max QoE = &\left(\prod_{i=1}^{N} QoE_i^{\frac{1}{N}}\right) \\ s.t. q_i(k) \in Q, \forall i = 1 \cdots N, \forall k = 1 \cdots K, \\ C_i(t) \leq & C_{BL}(t) \end{aligned} \tag{3}$$

Here, $C_i(t)$ is the bandwidth occupied by client i at time t, and $C_{BL}(t)$ is the size of the bottleneck bandwidth in the network at time t. Therefore, the constraint is that the bandwidth occupied by each client cannot exceed the bottleneck bandwidth on the network at any time. $Q$ is the set of all available video bitrate levels.

As it is uneconomical and complex to set up a centralized architecture for DAS over NDN which adopts a pull-based transmission paradigm, each client cannot know the value of the bottleneck bandwidth, nor can they know the total number of clients watching the same video through the bottleneck link. Therefore, the solution to the optimization problem of multi-client adaptive video streaming in NDN cannot be obtained by directly solving Eq.(3).

In this paper, we will design a distributed client-side bitrate adaptive algorithm that can improve the geometric mean of QoE of clients formulated as Eq.(3) without direct communication among the clients or scheduling by a central controller while improving network bandwidth utilization.

## IV. Reinforcement Learning Based Adaptive Bitrate Algorithm

In this paper, the Actor-Critic (AC) algorithm in reinforcement learning is used to construct the bitrate adaptive algorithm. And the AC algorithm is improved based on the policy gradient algorithm. We present a brief introduction to the Policy Gradient algorithm and the AC algorithm, respectively, in this chapter.

### A. Motivation

The scenarios considered in this paper are for multi-clients, where clients watch the same video, competing for the same bottleneck bandwidth. We expect to get the maximum average QoE, and simultaneously guarantee fairness between multiple clients, in the case of multi-clients. It is essential to obtain the maximum of average QoE that request aggregation between clients, in the case of individual clients. And when requesting aggregation between clients, the fairness between clients can also be improved. Because clients are not aware of the bottleneck bandwidth and information about other clients in the network, aggregation between clients is difficult. We introduce reinforcement learning methods where, through online training, client is able to obtain some information in the network environment, and then predict some information such as bottleneck bandwidth, which helps clients actively perform request aggregation.

### B. Policy Gradient Algorithm in Reinforcement Learning

The policy gradient algorithm seeks the optimal gradient with the gradient rise, for which an optimization objective is required. When optimization objective is the average reward for each step:

$$J_{avR}(\theta) = \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(s,a) R_s^a \tag{4}$$

Here, $\theta$ is the parameter of the policy, $d_{\pi_\theta}(s)$ is the static distribution of Markov chains based on the policy $\pi_\theta$. $R$ is the reward, $s$ is the status, and $a$ is the action.The policy gradient can be solved using a likelihood ratio method.

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(s,a) R_s^a \\ &= \sum_s d_{\pi_\theta}(s) \nabla_\theta \sum_a \pi_\theta(s,a) R_s^a \\ &= \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(s,a) \frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)} R_s^a \\ &= \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(s,a) \nabla_\theta \ln \pi_\theta(s,a) R_s^a \\ &= E\left[\nabla_\theta \ln \pi_\theta(s,a) R_s^a\right] \end{aligned} \tag{5}$$

For policy functions, the most commonly used is the softmax policy functions, which apply mainly for discrete space. The Softmax policy uses a linear combination of states of neural network output $(\Phi(s,a)^T \theta)$ to measure the probability of an action, namely:

$$\pi_\theta(s,a) = \frac{e^{\Phi(s,a)^T \theta}}{\sum_{a'} e^{\Phi(s,a')^T \theta}} \tag{6}$$

The corresponding score function can be obtained by derivation:

$$\nabla_\theta \ln \pi_\theta(s,a) = \Phi(s,a) - E\left[\Phi(s,\cdot)\right] \tag{7}$$

In (5), the partial derivative can be obtained by the mean of $\nabla_\theta \ln \pi_\theta(s,a) R_s^a$. The $\nabla_\theta \ln \pi_\theta(s,a) R_s^a$ is divided into two parts: $\nabla_\theta \ln \pi_\theta(s,a)$ and $R_s^a$. Where $\nabla_\theta \ln \pi_\theta(s,a)$ is a directional vector, called a score function, it represents the fastest changing direction of the $\ln \pi_\theta(s,a)$. The $R_s^a$ is a scalar, representing the amplitude. Both of them guide the update of the policy parameters.

In updates to the policy parameters, $R_s^a$ is like a critic, evaluating how much amplitude the policy parameters should be updated in a certain direction. Schulman J's paper [9] states that critics can have other forms such as TD error, advantage functions, action value, etc. And the policy gradient algorithm evolves Actor-Critic when critics represented in these new forms. In practice, the policy and evaluation functions in the

AC algorithm are usually expressed in two neural networks, called the Actor and Critic networks, respectively. When Critic is a TD error and is solved using the timing difference method,

$$\delta(t) = R_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{8}$$

Here $\gamma$ is the attenuation coeffcient. The policy network parameter update formula for Actor is:

$$\theta = \theta + \alpha \nabla_\theta \ln \pi_\theta(s,a) \delta(t) \tag{9}$$

The parameter update formula for the Critic network is:

$$\omega = \omega + \beta \delta(t) \Phi(s,a) \tag{10}$$

### C. AC Algorithm in Reinforcement Learning

The reason why reinforcement learning is chosen is that: the goal of reinforcement learning is to learn a strategy, through which the optimal action can be made to maximize the benefits. This is very similar to the model of the adaptive control algorithm: we expect to adaptively select the appropriate video bitrate, enabling the user to reach the maximum QoE. The currently selected bitrate also affects the later acquired QoE, which is consistent with the delay gain in reinforcement learning. Furthermore, reinforcement learning is able to obtain various information (such as the decision information of other clients) in constant interaction with the environment and take the benefits of bitrate aggregation into account, which can improve the fairness between clients. The specific process is as follows: the client selects a bitrate according to the initial policy and sends the bitrate request to the network; then the network returns the response data to the client, so that the client can observe the current QoE, and use it as a reward to update policy. After a period of time, the constantly updated and optimized strategy becomes the optimal strategy, guiding the client to obtain the maximum QoE, this is the learning process. Moreover, reinforcement learning is often used to solve discrete control problems and there are many mature framework models, which help the research in this paper.

The AC algorithm contains two parts, both Actor and Critic neural networks. Both of them have the same status input information. These information is the current buffer size, the download bandwidth of the last few segments, the download time of the last segment, the bitrate selection of the previous segment, and the number of segments remaining. The output of the Actor neural network, policy $\pi_\theta(s,a)$, is a probability vector representing the probability of selecting each bitrate. The Critic neural network outputs $V_\pi(s)$, is the evaluation of the current status. Because we wanted to get the maximum QoE, we chose QoE as the reward.

As described earlier, the AC algorithm is a policy gradient algorithm, critical on how to compute the gradient values of the expected cumulative returns regarding the neural network parameters. In the AC algorithm, we calculate $\delta(t)$ to replace $R_s^a$ , with (8), and then calculate the gradient with (5). The meaning of $\delta(t)$ is the difference in value before and after performing the action, which updating the parameters. Therefore, the parameter update formula for the Actor network is:

$$\theta = \theta + \alpha \nabla_\theta \ln \pi_\theta(s,a) \delta(t) \tag{11}$$

The Critic network outputs the $V_\pi(s)$, evaluating of the current status value. The parameter update formula for the Critic network is:

$$\omega = \omega + \beta \nabla_\omega \left( R_{t+1} + \gamma V(s_{t+1}) - V(s_t) \right)^2 \tag{12}$$

Here, the $R_{t+1} + \gamma V(s_{t+1}) - V(s_t)$ represents the difference between the actual value (the estimated target of the Critic network) and the estimated value. Since the actual value of the state $s_{t+1}$ cannot be directly calculated, the estimated value $V(s_{t+1})$ is used instead of the actual value. Guiding network parameter updates by difference between actual value and estimation value to make the state value estimation of the Critic network more accurate.

Due to the insufficient randomness of the AC algorithm itself, it is possible to fall into locally optimal intervals without reaching the global optimum. Therefore, this paper considers the introduction of exploration ideas for the agent to perform the selected actions randomly with a certain probability, so as to jump out of the local optimum interval and reach the global optimum.

### D. Training Setting

This paper has introduced a neural network model of the AC algorithm. In order to generate the AC algorithm model, the neural network in this paper employs a single hidden layer fully connected neural network. The training parameters are set as in Table 1.

The process trained in this paper is conducted under the deep learning algorithm library TensorFlow [10], which makes it easier to experiment.

TABLE I
TRAINING PARAMETER SETTINGS

| | |
|---|---|
| **State** | buffer size, download bandwidth of the last 5 segments, download time and bitrate of the previous segment, number of remaining video segments |
| **Action** | 20 bitrate selections |
| **Actor Network** | 9 input, 70 hidden and 20 output nodes |
| **Critic Network** | 9 input, 20 hidden and 1 output nodes |
| **Attenuation Coeffcient** | $\gamma = 1$ |
| **Degree of Exploration** | $\varepsilon = 0.3$ |
| **Training Rate** | Actor Network: $10^{-4}$<br>Critic Network: $10^{-3}$ |

## V. EXPERIMENTAL AND ANALYSIS

### A. Experimental Platform Introduction

Experiments in this paper use libdash [11] to build dynamic adaptive streaming platform in real-world environments. Libdash is a code framework that implements the DASH protocol standard, with a good interface for the bitrate control algorithm. Thus, we can add our algorithm to the code while compare the performance of other algorithms. Finally,

libdash contains a visual video player with intuitive algorithm performance. Therefore, this paper uses libdash to build a platform for real experiments.

The video used in the experiment was Big Buck Bunny[12]. This video lasts 10 minutes and has 299 clips, each lasting 2 seconds. Each segment has 20 different quality levels. Video is saved on the server side, while this article uses the Linux Traffic Control tool to limit the bottleneck bandwidth between the server and the router.Each node runs under the Ubuntu Linux 16.04 system with the same configuration. The individual nodes are connected together via the NFD. The experimental topological graph is shown in Fig. 2.

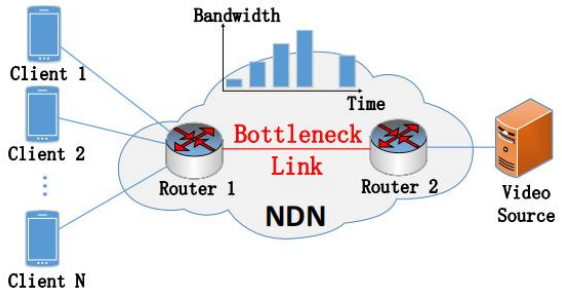


Fig. 2. Experimental topological graph.

### B. Comparative Algorithm and Performance Metrics

In this paper, the bitrate adaptive algorithm of Rate Based (RB), Rate&Buffer Based (BB), Game Based (GB)[13] are compared with the bitrate adaptive algorithm of Reinforcement Learning Based (RL) proposed in this paper. The performance of several algorithms are evaluated separately. The specific algorithms are detailed as follows:

- **RB:** This algorithm is mainly decided by using the bandwidth information with the past.
- **BB:** This algorithm utilizes both past bandwidth information and client buffer information for decisions.
- **GB:** This algorithm models the multi-client as a game theory model, designs a set of utility functions,and combines with the bandwidth information and buffer information in the past, to make the bitrate decision.
- **RL:** This algorithm is proposed in this paper. First it trains a neural network model, and then the bitrate decision can be made by inputting the collected state information such as buffer information into the neural network.

This paper selects the following performance metrics to evaluate the performance of the algorithm:

- **QoE:** the calculation method is as follows, including three parts: video quality, quality switching and interruption time. in this paper, we set$\gamma = 0.6, \alpha = 1, \beta = 4.3$

$$QoE = \sum_{n=1}^{N} q_n^{\gamma} - \alpha \sum_{n=2}^{N} \frac{|q_n - q_{n-1}|}{q_n} - \beta \sum_{n=1}^{N} T_n \tag{13}$$

- **Jain's fairness index:** The Jain's fairness index [14] defined as (14) is used to measure whether users or applications in the network share system resources fairly, the closer the value is to 1, the fairer it is:

$$J(x) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2} \tag{14}$$

- **Aggregation Ratio:** The ubiquitous caching in the NDN network enables aggregation of interest. Multi-clients can reduce redundant data and improve transfer efficiency.The calculation method is as follows:

$$Ar = \frac{agg}{N} \tag{15}$$

Where $agg$ is the number of segments being aggregated, and $N$ is the total number of segments.

### C. Experimental Analysis

In this paper, we experiment in all under two experimental scenarios. In these experimental scenarios, multiple clients want to watch the same video through a bottleneck bandwidth. In order to be more realistic, the bottleneck bandwidth of one of the experimental scenarios fluctuates.

- **Experimental scenario 1:** There are four clients, and the bottleneck bandwidth is limited to 6Mbps. The experimental results are shown in Fig. 3.
- **Experimental scenario 2:** There are four clients, and the bottleneck bandwidth is limited to 8Mbps (average), but there are fluctuations. The experimental results are shown in Fig. 4.

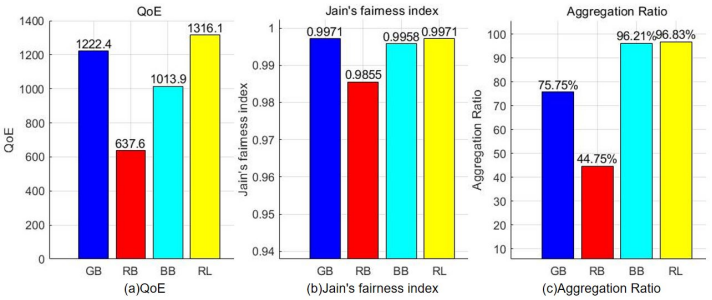


Fig. 3. Results of performance metrics for different algorithms (Experimental scene 1).

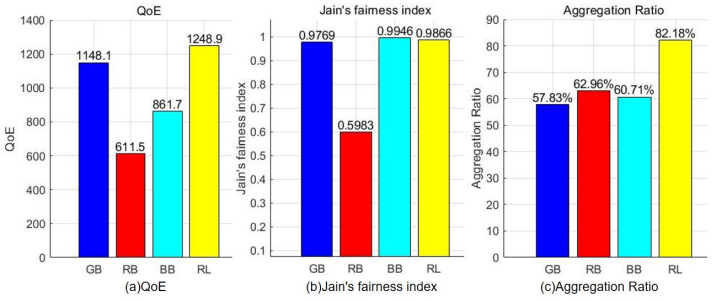


Fig. 4. Results of performance metrics for different algorithms (Experimental scene 2).

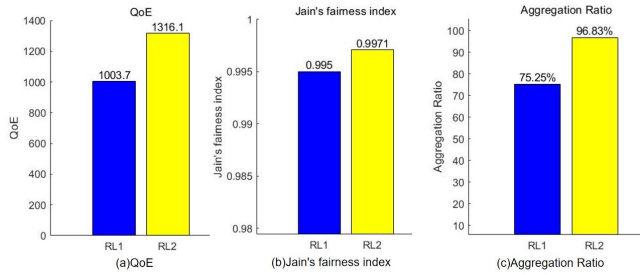


Fig. 5. Results of performance metrics for different algorithms (Experimental scene 2).

Through an analysis of the above data, it is found that our RL algorithm performs best on the QoE metrics. For other algorithms, the disadvantage of the RB algorithm is that only bandwidth is considered, which is prone to interruptions when the bandwidth changes and affecting the QoE. The disadvantage of the BB algorithm is its bitrate in a frequent change according to the buffer. The performance of GB algorithm is closest to our algorithm, which considers many factors such as buffer and bandwidth, but the utility function of GB algorithm is designed for experience and is not adjusted according to different scenarios. While our algorithm obtains the information in the scene from the training, and it constantly adjusts our strategy according to this information.

By comparing the two scenarios, we can find that the performance of all algorithms in scene 2 decreases due to bottleneck bandwidth fluctuations, but our algorithm reduces the least because it can learning and adaptive adjustment through training.

Furthermore, we compared the effect of online training on the experimental results of the RL algorithm, as shown in Fig. 5. Among them, 'RL2' is the experimental results obtained from repeated online training on the basis of 'RL1'. It can be seen that through online training, the RL algorithm can obtain various information in the experimental scenarios, resulting in better performance.

### D. Summary

By experiments, we found that the RL algorithm performs well due considering various factors such as buffer and bandwidth. At the same time, the RL algorithm is able to maintain superior performance in various environments because it is able to obtain information in the scene through constant training and adjust the strategy. And the information obtained by the client also contains the decision information of other clients, which helps to actively achieve bitrate aggregation among clients.

## VI. Conclusion

In this paper, we propose a multi-client dynamic adaptive video streaming algorithm called RL. We introduce reinforcement learning into the adaptive video stream of NDN multi-client scenarios. The proposed RL algorithm can achieve the active aggregation of requests between clients as much as possible without direct communication or scheduling with the central controller. We build an experimental platform and perform comparative experiments and evaluation of the proposed algorithm. Experimental results show that the RL algorithm has a higher performance in terms of overall QoE, fairness, and aggregate rate as compared to the other three comparison algorithms. But to simplify the complexity, the neural networks we used is simple three-layer fully connected neural networks. Using more efficient networks such as LSTM may provide better performance to the algorithm.

## Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2020YFA0711400, in part of Key Science and Technology Project of Anhui under Grant 202103a05020007, in part by the National Science Foundation of China under Grant 61673360, in part by the Key R&D Program of Anhui Province in 2020 under Grant 202004a05020078.

## References

[1] C. W. Paper, "Cisco visual networking index: Forecast and trends, 2017-2022," 2018.
[2] L. Zhang, A. Afanasyev, J. Burke, et al., "Named Data Networking," Comput. Commun. Rev., vol. 44, no. 3, pp. 66-73, 2014.
[3] Stockhammer T. Dynamic adaptive streaming over HTTP –: standards and design principles [C]. ACM Conference on Multimedia Systems. 2011: 133–144.
[4] NDN Team. NDN Forwarding Daemon, accessed on 2017. [Online].Available: http://named-data.net/doc/NFD.
[5] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A ControlTheoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in Proc.ACM SIGCOMM, 2015.
[6] S. Awiphan, K. Poobai, K. Kanai and J. Katto, "Proactive Interest Adaptation and Content Caching for Adaptive Bit-rate Video Streaming over NDN," in Proc. Int. Conf. Computer Commun. Syst., 2018, pp. 291-296.
[7] Z. Akhtar, YS. Nam, R. Govindan, et al., "Oboe: auto-tuning video ABR algorithms to network conditions," in Proc. ACM SIGCOMM, 2018.
[8] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in Proc. ACM SIGCOMM, 2017.
[9] Li W, Oteafy S M, Hassanein H S. Dynamic adaptive streaming over popularity-driven caching in information-centric networks[C]. Communications (ICC), 2015 IEEE International.Conference on. : IEEE, 2015:5747–5752.
[10] M. Abadi et al. 2016. TensorFlow: A System for Large-scale Machine Learning. In OSDI. USENIX Association.
[11] Mueller C, Lederer S, Poecher J, et al. Demo paper: Libdash - an open source software library for the MPEG-DASH standard[C].IEEE International Conference on Multimedia and Expo Workshops. 2013: 1–2.
[12] Lederer S, Timmerer C. Dynamic adaptive streaming over HTTP dataset[C].ACM Conference on Multimedia Systems. 2012: 89–94.
[13] X. Tan, L. Xu, J. Ni, S. Li, X. Jiang and Q. Zheng, "Game Theory Based Dynamic Adaptive Video Streaming for Multi-client over NDN," in IEEE Transactions on Multimedia, doi:10.1109/TMM.2021.3100768.
[14] Jain R, Chiu D, Hawe W. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems[R]. Technical Report TR-301, 1984.
[15] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal Bitrate Adaptation for Online Videos," in Proc. IEEE INFOCOM, 2016.
[16] I. Irondi, Q. Wang, C. Grecos, et al., "Efficient QoE-Aware Scheme for Video Qality Switching Operations in Dynamic Adaptive Streaming," ACM Trans. Multimed. Comput. Commun. Appl., vol. 15, no. 1, article. 17, 2019.
[17] A. Elgabli, V. Aggarwal, S. Hao, et al., "LBP: Robust rate adaptation algorithm for SVC video streaming," IEEE/ACM Trans. Netw., vol. 26, no. 4, pp. 1633-1645, 2018.