This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3100768, IEEE Transactions on Multimedia

Game Theory Based Dynamic Adaptive Video Streaming for Multi-client over NDN

Xiaobin Tan, Lei Xu, Jiawei Ni, Simin Li, Xiaofeng Jiang and Quan Zheng

***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 game theory based (GB) 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 between a DAS client and network as an incomplete information non-cooperative game. Then, this game is transformed into a complete but imperfect information game by Harsanyi transformation, and each client can issue an appropriate bitrate request by solving the Bayesian Nash Equilibrium (BNE) problem respectively. By designing the payoff function pair elaborately, the equilibrium point of the game can correspond to the situation that multiple clients issuing the same video bitrate request, that is, requests aggregation, which will reduce the repeated traffic and also achieve fairness. Compared with the existing solutions, through simulation and real-world experiments in multi-client video distribution scenarios, the GB algorithm outperforms the comparison algorithms in terms of overall Quality of Experience (QoE), fairness, and network bandwidth utilization, etc.**

***Index Terms*—Named Data Networking, Dynamic Adaptive Streaming, Multi-client, Game Theory.**

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. Furthermore, a large amount of redundant or repetitive traffic is transmitted on the Internet when a large number of users watch the same video, which leads to the inefficiency of network bandwidth resource utilization and unfairness between users. However, with the TCP/IP protocol as the core, current Internet architecture only cares about the source and destination address of the data, but not the content itself, so it cannot effectively solve the problem of redundant transmission. Whereas, Named Data Networking (NDN) [2], centering on content instead of the address, can reduce redundant network traffic dramatically, especially under the content distribution scenario [12][13][14][20][40].

NDN is a clean-slate future Internet architecture that shifts the host-centric communication model to content-centric model supporting among unique content names, in-network caching, and name-based routing. NDN can reduce redundant network traffic of content distributions by the ability of content aggregation, which means the request from multiple clients for the same content only needs to be replied with one piece of content on the network links instead of corresponding multiple copies. There are two types of aggregation in NDN, Pending Interest Table (PIT) aggregation and Content Store (CS) aggregation. The PIT aggregation means that when an NDN node receives multiple *Interests* packets with the same name, it may fill them in one PIT entry instead of multiple entries, and then forward the *Interests* packet to the upstream node only once. Then the *Data* packets with the same name will only be transmitted once on the link, thereby greatly reducing network bandwidth consumption. While the CS aggregation means that the NDN nodes can directly return the requested content cached in local CS without forwarding the request to the data source.

It is worth mentioning that whether deployed on the edge of the network or inside the network, NDN nodes have the capability of aggregation. In the broad scenarios, whose network structure is often a tree-like hierarchical structure, the resource bottleneck phenomenon is more obvious at the higher-level network links. Each NDN nodes have the capability of aggregation if it receives multiple *Interest* packets with the same name whether deployed on the edge or inside the networks. So our proposed GB algorithm is also applicable for broad NDN networks scenarios, not just home networks. In addition, from the perspective of video playback mode, real-time TV broadcasting is the main scenario we consider because it is an important type of video service scenario. More than that, even in video-on-demand scenarios, for some hot videos, there may also exist some concurrent requests for an identical video from multiple users, and if these requests are aggregated, the traffic of duplicate data will be reduced.

As a popular technology used for video distribution applications on the Internet, Dynamic Adaptive Streaming (DAS) can be deployed on NDN and can improve its performance by leveraging the aggregation capability of NDN. The pioneering NDN work [2] and some research results [3] demonstrates that NDN advantage against TCP/IP in the content distribution scenario. The reason is that NDN can reduce redundant network traffic of content distribution by the ability of content aggregation, which means the request from multiple clients for the same content only needs to be replied with one piece of content on the network links instead of corresponding multiple copies. Through comprehensively analyzing, we consider that the essential difference between DAS in the TCP/IP network and DAS in NDN is that there exists not only bandwidth

Xiaobin Tan, Lei Xu, Jiawei Ni, Simin Li, Xiaofeng Jiang and Quan Zheng are with Laboratory for Future Networks, University of Science and Technology of China, Hefei, China, 230027.
Lei Xu is the corresponding author (e-mail: xl10092@mail.ustc.edu.cn).
E-mail: {xbtan@, jxf@, qzheng@}ustc.edu.cn, {nijiawei@, lisimin@}mail.ustc.edu.cn

competition but also the possibility of collaboration (content aggregation) among clients in NDN. Therefore, the adaptive bitrate (ABR) algorithm designed for multi-client scenarios over NDN should take into full consideration the aggregation capability of NDN to improve the performance of video distribution in NDN.

Existing research on client-based DAS over NDN only considers bandwidth competition among clients rather than aggregation ability of NDN, which leads to unsatisfactory performance of video distribution services in multi-client scenario suffered from low aggregation ratio. Therefore, the aggregation ability of NDN should be fully considered to further improve the performance of DAS in NDN. Besides, since the content may be fetched from different cache nodes in NDN, it is difficult or unrealistic to coordinate clients' behavior through centralized control or direct communication among clients. That is to say, bitrate adaptive of DAS over NDN can be regarded as a distributed collaborative optimization problem. Therefore, we need to design a fully client-side ABR algorithm over NDN that can achieve an optimal overall quality of experience (QoE) of multiple clients and guarantee their fairness by proactive aggregation among clients as much as possible. Here, QoE fairness refers to the fairness of QoE among the clients watching the same video. In this paper, we adopt the Jain's Fairness Index [7] as the metric of fairness for users' QoE. It is a well-accepted metric that measures whether users or applications in the network share system resources fairly while not unduly sensitive to atypical network flow patterns.

In this paper, we propose a distributed ABR algorithm for NDN based on game theory because it is efficient in solving the distributed collaborative optimization problem. By modeling the process of bitrate adaptive in NDN as a non-cooperative incomplete information game, i.e. Bayesian game, we propose a distributed bitrate adaptive algorithm based on the process of solving for the Bayesian game. In this game, a single client is regarded as one *player*, and the network and the rest of the clients is another *player*. The client can request a video segment with the bitrate corresponding to the Bayesian Nash Equilibrium (BNE) point according to the appropriate payoff function pair we designed, which synthetically considers the factors such as available bandwidth, buffer occupancy, video bitrate, etc. To our knowledge, this is the first ABR algorithm that considers proactive bitrate request aggregation in a multi-client DAS scenario over NDN.

The main contributions of this paper are as follows:

- **Model the multi-client DAS over NDN as a Bayesian game**: We model the multi-client DAS over NDN as a Bayesian game and design a payoff function pair for *players*. Then, this game is transformed into a complete but imperfect information game by Harsanyi transformation, and each client can issue an appropriate bitrate request by solving the BNE problem.
- **Client-side ABR algorithm for NDN based on game theory**: We propose a novel client-side ABR algorithm based on game theory that can achieve proactive aggregation among clients as much as possible without requiring coordinating with other clients or scheduling by a central controller. The proposed algorithm can optimize the overall QoE of multiple clients and guarantee their fairness while reducing the consumption of network bandwidth.
- **Prototype developed on realistic NDN platform**: We evaluate the proposed algorithm on ndnSIM, a simulator for NDN. And we develop a dynamic adaptive video streaming system over NDN (DAS-NDN) based on Libdash. Experimental results on simulation and realistic platform show that the GB algorithm outperforms the comparison algorithms in terms of overall QoE, fairness, and network bandwidth utilization in the multi-client DAS scenario of NDN.

The rest of this paper is organized as follows. In Section II, the related work on the single-client and multi-client adaptive video streaming transmission scheme for NDN is presented mainly. In Section III, the system architecture and QoE model are established and the problem is described. In Section IV, we build a Bayesian game model for multi-client DAS over NDN and propose a novel client-side ABR algorithm based on game theory. Simulation and real-world experimental results are given in Section V and VI, respectively. Finally, in Section VII, we summarize the work of this paper and make further prospects for future research work.

II. Related Work

Existing ABR algorithms in NDN can be classified into two categories based on design objective, single-client algorithm, and multi-client algorithm. If an algorithm only makes bitrate decisions based on the current client itself, it is a single-client algorithm, while if an algorithm considers the QoE of all clients and the utility of network resources, it is a multi-client algorithm. Because the client-side algorithms designed for HTTP can be mitigated to NDN by being slightly modified, we also introduce some related researches for HTTP.

A. Single-client Adaptive Video Streaming

There are not so many single-client ABR algorithms specifically designed for NDN. Liu et al. [26] introduced a hop-by-hop adaptive video streaming scheme called HAVS-CCN. They encode the video into multiple layers with different bitrates based on H.264/SVC, and the node drops some enhancement layers, sacrificing a certain amount of video quality in exchange for smooth video playback when the network is congested and the link capacity cannot maintain the requested video quality. Awiphan et al. [27] made further improvements on HAVS-CCN to improve the performance of rate adaptation and video streaming. They designed an algorithm that allows *Interest* adaptation in a hop-by-hop way along with the proactive content caching. However, these schemes will bring extra overhead to network nodes.

There are some typical algorithms in HTTP that are purely client-side based [10], [29]-[31], [33], [34], 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3100768, IEEE Transactions on Multimedia

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

To the best of our knowledge, there exists only one multi-client DAS algorithm designed for NDN. Alt et al. [28] formulated the quality adaptation decision for QoE maximization in ABR video streaming as a contextual multi-armed bandit problem. They proposed a sparse Bayesian contextual bandit algorithm, which can provide real-world video players with quality adaptation decisions based on the network context. Although they considered the issue of fairness among multiple clients, they did not consider the aggregation capability of NDN in the multi-client scenario, which may lead to low bandwidth utilization. Moreover, they modified the architecture of the NDN.

In the TCP/IP based network, many researchers [13], [17]-[19] proposed some schemes by controlling the download rate of the video segments to achieve relative fairness among multiple clients. Recently, some researches studied the bitrate adaptive for DASH based on cooperative game [25], [35]-[37], or non-cooperative game [32] models. The approaches based on cooperative game theory rely on centralized control, which is difficult to be mitigated in NDN. Based on non-cooperative game theory, [32] proposed an algorithm that can optimally allocate the server's limited export bandwidth to multiple clients. However, in their proposed solutions, an additional HTTP session is required to be maintained between the client and the server, which is also impossible in NDN.

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.

III. System Modeling and Problem Formulation

A. System Architecture

The architecture of dynamic adaptive streaming over NDN is shown in Fig.1. The video file is encoded into fixed-duration segments that are made available at the video server with multiple bitrates, and the client adaptively requests video segments of appropriate bitrate aiming at maximizing his watching experience.

Firstly, the client requests a manifest file called Media Presentation Description (MPD) which containing information and metadata about this video. The client then requests video segments of appropriate bitrate instructed by bitrate controller module based on the information such as network states and buffer size, etc. Finally, the requested video segments are transmitted over the NDN network via NDN Forwarding Daemon (NFD) [4] to the local video buffer, and taken out when playback.

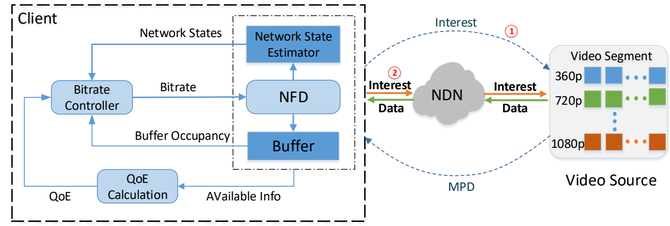


Fig. 1: Architecture of dynamic adaptive streaming over NDN.

It should be highlighted that in multi-client scenario of DAS over NDN, which is different from DAS over TCP/IP, there is not only competition in bandwidth but also cooperation in bitrate aggregation among the clients. The reason is that, in NDN, if multiple clients request the same bitrate version of the same video segment simultaneously, only one copy of the video segment will be transmitted, which will reduce the transmission of redundant data dramatically. However, since it is difficult for clients to communicate with each other directly or scheduling by a central controller in DAS over NDN, we hope to design a distributed ABR algorithm that can improve the overall QoE of clients by increasing the aggregation probability of video requests issued by them. Besides, the proposed ABR algorithm should ensure fairness among the clients as much as possible.

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 [33], which is widely accepted and used in the literatures such as [27][30][31]. The QoE metric is a linear combination of the following three key components:

Video quality: The relationship between bitrate level and QoE are positively correlated, but not linearly proportional. In the case of a relatively low video bitrate level, a small increase in bitrate level will bring a considerable improvement in QoE. Whereas, the increase in bitrate level may not have a significant improvement for the user's QoE in the case of a high bitrate level. Consequently, we define the video quality as q^r, where q is the bitrate level of the video segment and $r \in (0, 1]$ is a constant. According to [38], we set the default value of r to 0.6.

Quality switching: This component measures the consistency of video quality during playback and is defined as the difference in video quality between two consecutive video segments $\left|\frac{q-\bar{q}}{q}\right|$, where $\bar{q}$ is the bitrate level of the previous video segment. After designing in this fashion, we can ensure that the switching of video quality from high to low will has a greater impact than that of switching from low to high. In other words, this design is consistent with people's preference for higher video quality.

Interruption time: The total duration of buffered video segments was used to represent the size of the buffer. The video playback will be paused if the requested segment can't arrive before the buffer is empty, which we call interruption. The interruption time can be formulated as $\max(\frac{size(q)}{b} - B, 0)$

with $size(q)$ being the size of the selected segment, B being the current size of the buffer and b being the bandwidth it takes up to download it.

Then, the QoE for a specific client when watching a video segment with bitrate level q can be formulated as following:

$$qoe = q^r - \alpha\left|\frac{q-\bar{q}}{q}\right| - \beta\max(\frac{size(q)}{b} - B, 0). \quad (1)$$

Here, the α and β are non-negative weighting parameters corresponding to the video quality switching and interruption time, respectively. Video viewers can set these two parameters according to their preferences. A relatively large α indicates that the viewer is particularly concerned about video quality switching. A relatively large β indicates that a viewer is very concerned about interruption time.

Furthermore, we define QoE of the i-th client during the whole video (with a total of K segments) playback process as:

$$\begin{aligned} QoE_i = &\sum_{k=1}^{K} q_i(k)^r - \alpha_i \sum_{k=2}^{K}\left|\frac{q_i(k)-q_i(k-1)}{q_i(k)}\right| \\ &- \beta_i \sum_{k=1}^{K}\max(\frac{size(q_i(k))}{b_i(k)} - B_i(k), 0), \end{aligned} \quad (2)$$

where $q_i(k)$ and $size(q_i(k))$ denote the bitrate level and actual size of the k-th segment obtained by the i-th user, respectively. $B_i(k)$ and $b_i(k)$ denote the size of the buffer and the bandwidth occupied when the i-th user downloaded the k-th segment, respectively.

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

$$\begin{aligned} &\max_{q_i(k)} QoE = (\prod_{i=1}^{N} QoE_i)^{\frac{1}{N}}, \\ &s.t.\ q_i(k) \in Q,\ \forall i = 1\cdots N,\ \forall k = 1\cdots K, \\ &\qquad C_i(t) \le C_{BL}(t). \end{aligned} \quad (3)$$

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.

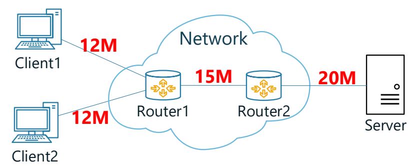


Fig. 2: Multi-client DAS scenario: Two clients want to watch the same video. The numbers in red indicate the maximum limited bandwidth of corresponding links, the same below.

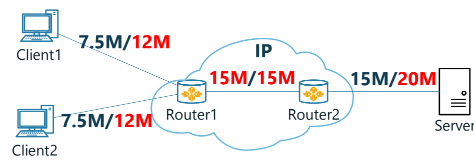


Fig. 3: Multi-client DAS over the TCP/IP based network. The numbers in blue indicate the occupied bandwidth on corresponding links, the same below. The reasonable result that can be achieved in the end is two clients each obtains half of the bottleneck bandwidth.

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. Game Theory Based Adaptive Bitrate Algorithm

A. Motivation

In order to explain our idea more clearly, we illustrate the difference between multi-client DAS in NDN and TCP/IP using a simple example. Considering a scenario shown in Fig.2, two clients are watching the same video clip through a shared bottleneck link. The reasonable result of multi-client video distribution service in TCP/IP network is that the two clients get 7.5Mbps network bandwidth respectively as shown in Fig.3, and each client can watch the video with bitrate below 7.5Mbps.

As a contrast, multi-client video distribution service in the NDN network can improve the QoE for all users who are

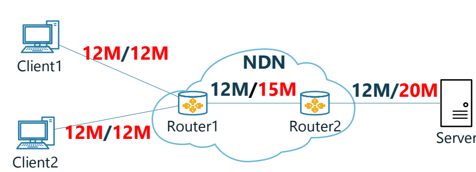


Fig. 4: Multi-client DAS over the NDN. The ideal result is that the requests of video segment bitrate from the two client requests are aggregated and then both clients occupy 12M network bandwidth.

watching the same video segment by leveraging the capability of aggregation as shown in Fig.4. In this scenario, if client 1 and client 2 request the same bitrate version of the same video segment, only one copy of the video segments will be transmitted on the bottleneck link. Under ideal conditions, all clients can watch video segments with higher bitrate simultaneously. Moreover, this also guarantees the fairness of video distribution service among multiple clients because they watch the same bitrate version of the video.

It can be seen that under ideal conditions, the multi-client DAS in NDN can not only achieve fairness among multiple clients but also improve the utilization of network bandwidth resources, thereby improving the satisfaction of all clients. The main reason is that DAS in NDN can leverage the capability of aggregation to reduce redundant traffic in the network drastically and then improve the efficiency of video distribution service.

Since the video content may be fetched from different cache nodes in NDN, it is difficult or unrealistically to coordinate clients' behavior through centralized control or direct communication among clients in DAS over NDN, we hope to design a distributed ABR algorithm which can improve the QoE of clients by increasing the aggregation probability of video requests issued by them. In this paper, we will design a distributed ABR algorithm based game theory, which is a mathematical theory that is capable of solving a distributed collaborative optimization problem.

B. Non-cooperation Game model for DAS over NDN

Game theory studies situations in which many $players$ make decisions independently, and it can be classed into cooperation game and non-cooperation game. In this paper, we want to design a distributed 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. Non-cooperative game is a feasible solution to accomplish this goal.

Therefore, we model the bitrate adaptive for multi-client DAS over NDN using the non-cooperation game theory. In this scenario, each client neither knows the bandwidth of the bottleneck link nor the number of users who watch the same video through this bottleneck link, so each client can only make bitrate decisions according to its state. According to the classification of the non-cooperation game theory, this scenario can be modeled as a non-cooperative incomplete information game, i.e. Bayesian game. Furthermore, the distributed optimization of bitrate adaptive for multiple clients in NDN can be achieved by solving the BNE problem.

In this game model, we first need to determine the three elements of the game: *player*, *Strategy*, and *Payoff*. Through the above conversion method, we define *player 1* as a single client and *player 2* as the rest of the network. The strategy that *player 1* can choose is the available video bitrate that is described in the MPD file. The strategy that *player 2* can choose is the amount of bandwidth allocated to *player 1*.

Assuming that a *player* has a private message (not known to other *players*), then the different values of the private information are called different *types* [5] of the *player*. Since *player 1* is a single client, it is not aware of the network conditions, and its primary concern is the amount of bandwidth that other clients in the same network have already occupied, while this information is only known to *player 2*. We use w to represent the total bandwidth occupied by other clients in the same network, upon that the w is the *type* (a term of game theory) of *player 2*. Although the specific size of w is unknown to *player 1*, its prior probability distribution is known to subject to the Gaussian truncation distribution [5]. In each decision round, the client chooses a strategy from his strategy space that can maximize his payoff.

In a non-cooperative game, each client makes a decision only based on its payoff function. Therefore, the payoff function should be designed with the capability of encouraging the clients to issue the same bitrate request. *Player 1*'s payoff function should reflect its profit and the cost of obtaining the profit and can help the client to achieve higher cost-effective decisions. The most important goal of *player 2*'s payoff function is to achieve fair bandwidth allocation and then to achieve proactive aggregation of requests among clients. After comprehensively considering multiple influencing factors and conducting multiple experiments, the payoff functions we designed are as follows.

$$p_1(br\_q, b) = \frac{qoe}{1 + e^{\theta(br\_q - \lambda b)}}, \tag{4}$$

$$p_2(b, w) = -(1 - \frac{b}{w})^2. \tag{5}$$

Here, br_q denotes the bitrate of the video segment, b is the bandwidth allocated to *player 1* by *player 2*, w is the total bandwidth occupied by other clients. The θ is a weight parameter that balances the proportional relationship between gain and cost. The larger the θ, the higher the cost of choosing a high bitrate, indicating that the client's strategy at this time tends to be more conservative. The λ is the parameter related to the buffer status of the client and is generally set to 1.0, or 0.8 if buffer occupancy is less than 33%. From the definition of payoff function for *player 1*, the more the bitrate of the video segment exceeds the allocated bandwidth, the smaller the value of the payoff function will be.

For *player 1*'s payoff function Eq.(4), the numerator qoe (as shown in Eq.(1)) is his gain of watching the video, and the denominator is used to adjust the value of the payoff function according to the bitrate requested. The denominator is always larger than 0, which means that as long as the video segment is requested, there exists cost. If the requested bitrate level of the video segment is too high and exceeds the network bandwidth limit, the requested video segments will be delayed or even dropped, which will cause playback interruption. This form of the denominator is designed to penalize this behavior and to induce users not to request excessive bitrates level.

For *player 2*, since the optimal game result of the multi-client video service is that all clients get bandwidth fairly, that is, the aggregation of the requested content is fully achieved, and there will be $w = b$. Accordingly, the Eq.(5) is designed to allow *player 2* to reach its maximum value at $b = w$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3100768, IEEE Transactions on Multimedia

Through the Harsanyi transformation, we can transform incomplete information game into a complete but imperfect information game by introducing $Nature$ as a participant into the game [39], which can be analyzed by Bayesian Law and other probability theory. After that, the client can determine the bitrate of the video segment to be requested according to the BNE point of the game.

C. Solution of Bayesian Nash Equilibrium Problem for Bitrate Adaptive

We should analyze the payoff function pair of this game to solve the BNE problem. The real value of w is only known to *player 2* and his payoff gets maximum when $b = w$ which means *player 1* can share bandwidth fairly with the others at this moment. The w subjects to the Gaussian truncated distribution between a and A (the lower and upper limits of bottleneck link bandwidth, respectively), with average μ_w, variance σ_w^2 and probability density function $f_0(w)$:

$$f_0(w) = \frac{\frac{1}{\sigma_w}\phi(\frac{w-\mu_w}{\sigma_w})}{\Phi(\frac{A-\mu_w}{\sigma_w}) - \Phi(\frac{a-\mu_w}{\sigma_w})}. \tag{6}$$

Here, $\phi(\xi) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\xi}$ is the standard normal distribution and $\Phi(\xi) = \frac{1}{2}(1 + erf(\frac{\xi}{\sqrt{2}}))$ is its cumulative distribution function. Where $erf(x)$ is the error function as shown below:

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt. \tag{7}$$

In each decision round, the choice of *player 2* only depends on w and the optimal response corresponding to the selection of the most suitable b is denoted by s_2^*:

$$s_2^*(w) = \arg\max_b p_2(b, w) = w. \tag{8}$$

Simultaneously, the optimal response for *player 1* is denoted as follows:

$$s_1^*(w) = \arg\max_q p_1(q, w). \tag{9}$$

Now, as strategy s_1^* is the optimal response to s_2^* and knowing that s_2^* is the optimal response to any strategy by *player 1*, (s_1^*,s_2^*) is as a subgame-perfect equilibrium [5] for each decision round. The uncertainty on the *type* of *player 2* make *player 1*'s response sub-optimal. In the first decision round, *player 1* has no information on *player 2*'s *type*, and it chooses to act the BNE:

$$s_1^* = \arg\max_q \int_a^A p_1^*(q, \delta) f_0(\delta)\,d\delta, \tag{10}$$

where $f_0(\delta)$ is as shown in Eq.(6), and $p_1^*(q, \delta)$ is the maximum payoff that *player 1* can get at this time.

Because Eq.(9) is a nonlinear function, and the integral formula cannot be derived directly, the approximate solution of the integral can be calculated according to the Simpson's Rule [6]. In the process of repeating the game, the *player 1* can obtain the estimated value of w continuously and update the estimated distribution of w, so that the estimation of w is more accurate and the bitrate selection of the video segment is more reasonable.

Before each decision round, *player 1* gets some information about *player 2*'s *type* from the previous segment to modify the probability distribution of w. Thus, *player 1* can estimate b by the download speed of the previous segment which is revealed from the size and transmission time of the segment:

$$\hat{b} = \frac{size(q)}{T_2 - T_1}. \tag{11}$$

With T_1 being the time when the first *Interest* packet is sent and T_2 being the time when the last data packet is received. Thus, the estimated value of w is defined as follows:

$$\hat{w} = \hat{b}. \tag{12}$$

The process of the game is continually repeating and we can update the prior estimation for w after each decision round. After a sequence of n further refined estimates of the *type* $(\hat{w}_1, \hat{w}_2, ..., \hat{w}_n)$, we can update the prior estimate by using the mean and variance unbiased estimators:

$$\hat{m}_w = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_i, \tag{13}$$

$$\hat{v}_w^2 = \frac{1}{n-1}\sum_{i=1}^{n}|\hat{w}_i - \hat{m}_w|^2. \tag{14}$$

Then, we can estimate the new prior distribution parameters for w, average $\hat{\mu_w}$ and variance $\hat{\sigma_w^2}$ by (15) and (16):

$$\hat{\mu}_w = \hat{m}_w + \frac{\phi(x_1) - \phi(x_2)}{\Phi(x_2) - \Phi(x_1)}\hat{v}_w, \tag{15}$$

$$\hat{\sigma}_w^2 = \hat{v}_w^2[1 + \frac{x_1\phi(x_1) - x_2\phi(x_2)}{\Phi(x_2) - \Phi(x_1)} - (\frac{\phi(x_1) - \phi(x_2)}{\Phi(x_2) - \Phi(x_1)})^2]. \tag{16}$$

With $x_1 = \frac{a-\hat{m}_w}{\hat{v}_w}$, $x_2 = \frac{A-\hat{m}_w}{\hat{v}_w}$, $\phi(\xi)$ being the standard normal distribution and $\Phi(\xi)$ being its cumulative distribution function as mentioned above.

For the first segment, the prior estimate of w is set empirically in advance. Besides, the lower limit of it is always 0 and the upper limit is maximum bandwidth estimated. After the acquisition of update prior estimation, the algorithm will iterate all available bitrate figure out the payoff of *player 1* by Eq.(10) and choose to request the bitrate that can maximize payoff. Through the continuous update prior estimation, *player 1* will have more and more knowledge of *player 2* and choose an appropriate bitrate to request.

D. Algorithm Workflow

Based on the analysis of the gaming process, the workflow of the game theory-based (GB) ABR algorithm is shown in Algorithm 1.

Firstly, the client needs to get some basic information: B is the total duration of buffered video segments in the local buffer. b is the bandwidth allocated by the *player 2* to the *player 1*, that is, the average bandwidth during the download of the previous segment. C is the estimated bandwidth. $\bar{q}$ is the bitrate of the previous segment.

Before making a bitrate decision, the client updates the estimated distribution of w based on the previous request, as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3100768, IEEE Transactions on Multimedia

shown in lines 6-9. Next, client sets the value of λ based on the local buffer condition, as shown in lines 10-14, where B_{max} is the total size of the local buffer. All preparations are completed and the bitrate selection is now started, as shown in lines 15-23. Traverse for all available bitrates without abandon unless the selected bitrate brings interruption to the client. There are two purposes for this: 1. Preventing interruption which has a fatal blow to the client's QoE; 2. Excluding some bitrates that do not meet the requirements, and reducing the amount of calculation.

Although we can get the average bitrate of the video segment for each bitrate, the actual size values of them are not known until they are received by the client. For a video segment that requested by a client, the expectation value of the payoff is calculated according to Eq.(2). Finally, the client selects the bitrate that maximizes the expected value of the payoff.

Algorithm 1 Game Theory Based Adaptive Bitrate Algorithm

Require: $B, b, C, \bar{q}$
Ensure: $highest\_bitrate$
1: $B \leftarrow Get\_Buffer()$;
2: $C \leftarrow Get\_Estimate\_Bandwidth()$;
3: $b \leftarrow Get\_Last\_Download\_Speed()$;
4: $\bar{q} \leftarrow Get\_Last\_Download\_Bitrate()$;
5: $highest\_bitrate \leftarrow Get\_Lowest\_Bitrate()$;
6: $w \leftarrow b$;
7: $Set\_W(w)$;
8: $\hat{\mu}_n \leftarrow calculate\_average\_w()$;
9: $\hat{\sigma}_n^2 \leftarrow calculate\_var\_w()$;
10: **if** $B < 0.33 * B_{max}$ **then**
11: $\quad \lambda = 0.8$;
12: **else**
13: $\quad \lambda = 1.0$;
14: **end if**
15: **for** $allavailablebitrate$ **do**
16: $\quad$ **if** $2 * current\_bitrate/C < B$ **then**
17: $\quad\quad current\_p\_1 \leftarrow calculate\_p\_1()$;
18: $\quad\quad$ **if** $current\_p\_1 > p\_1$ **then**
19: $\quad\quad\quad p\_1 = current\_p\_1$;
20: $\quad\quad\quad highest\_bitrate = current\_bitrate$;
21: $\quad\quad$ **end if**
22: $\quad$ **end if**
23: **end for**

It can be seen from the Algorithm 1, for a single user to watch a single video segment, the complexity of the GB algorithm only depends on the number of candidate bitrate levels (indicated by N_{avl}). Therefore, the complexity can be expressed as $O(N_{avl})$. In fact, the N_{avl} is a small integer (usually less than 20 [24]) and the complexity of the GB algorithm is quite low, given the computing power of today's hardware devices, the delay caused by the GB algorithm's computing is negligible. It is worth mentioning that although the basic unit of data transmission in NDN is a chunk, and a video segment is composed of a sequence of consecutive chunks. When a user watches a video, the bitrate decision is made for each video segment, not for each chunk, so the ABR algorithm is run once per segment. Besides, since the proposed algorithm is a client-side adaptive bitrate algorithm, each client makes decisions only based on his state and network conditions. Therefore, the number of concurrent users will not affect the complexity of the proposed algorithm running on each client, even if the users are on different live channels.

V. Performance Evaluation for Simulation

Experiments on the simulation environment are flexible, which allows us to evaluate the proposed algorithm on all metrics conveniently. The simulation experiment in this section is carried out on ndnSIM [11] (an NS-3 based NDN simulator).

We compare the proposed algorithm with three classic ABR algorithms, namely Rate-Based (RB) [9], Buffer-Based (BB) [8], and Buffer Occupancy based Lyapunov Algorithm (BOLA) [10], whose data transmission methods are modified to be suitable for adaptive streaming in NDN. The adaptive bitrate algorithms work in the application layer, so they can run on different network architectures, such as NDN or TCP/IP. In the TCP/IP network, the clients only need to send an "HTTP GET" instruction to request the desired video segments. The actual data transmission process is undertaken by the TCP protocol. However, due to the lack of aggregation capabilities, the DAS running on the TCP/IP networks cannot aggregate usersąŕ requests, so there exists repeated data transmission, resulting in low system performance. While in the NDN network, which has no mature transport layer protocols, the transmission of video data needs to be implemented by applications itself. First, the provider named and published MPD files and video files that have been segmented according to NDN rules. A video client gets an MPD file by sending corresponding *Interests*, and it will get video relevant information by resolving this file. Then it sends out sequences of Interest packets according to the name and the number of video blocks. When the block ID in the returned *Data* packet matches the FinalBlockID field in its MetaInfo, the client can reassemble the received *Data* packets into a complete video segment for playback.

The reason for not comparing with existing ABR algorithms in NDN is that most of the existing algorithms modify the architecture of NDN more or less, which is inconsistent with our design philosophy. We perform simulations in the following two scenarios:

Scenario 1: Under the premise of fixed bottleneck bandwidth, multiple clients request the same video;

Scenario 2: Under the premise of fluctuating bottleneck bandwidth, multiple clients request the same video.

Finally, we introduce some performance metrics used in evaluating the performance of algorithms from different perspectives, such as the sum of all users' QoE, fairness, aggregation ratio, and bandwidth resource utilization.

A. Performance Metrics

We introduce several metrics for evaluating the performance of the bitrate adaptive algorithm.

- **Sum of All Users' QoE**: This performance metric can directly reflect the overall service quality of the DAS system, assuming a total of N clients, as shown in Eq.(17):

$$QoE_{sum} = \sum_{i=1}^{N} QoE_i, \tag{17}$$

where N represents the total number of clients watching video simultaneously.

- **Average Video Quality**: The arithmetic mean of the video quality watched by users during the video playback process, as shown in Eq.(18):

$$Q_{aq} = \frac{1}{K}\sum_{k=1}^{K} q(k)^r, \tag{18}$$

where $q(k)$ is represented by the index of the corresponding bitrate, and $r \in (0, 1]$ being a constant set to 0.6.

- **Average Video Quality Switching**: The average switching of video quality during video playback, as shown in Eq.(19):

$$Q_{qs} = \frac{1}{K-1}\sum_{k=2}^{K} |q(k) - q(k-1)|, \tag{19}$$

which reflects the smoothness of the video switching.

- **Average Interruption Time**: The average interruption time during video playback, as shown in Eq.(20), in seconds:

$$Q_{it} = \frac{1}{K}\sum_{k=1}^{K} \max(\frac{size(q(k))}{b(k)} - B(k), 0), \tag{20}$$

which reflects the stability of the video playback process.

- **Geometric Mean of QoE**: The geometric mean not only reflects the overall watching experience of all clients but also the difference among the clients, as shown in Eq.(21).

$$QoE_{gm}(k) = (\prod_{i=1}^{N} QoE_i(k))^{\frac{1}{N}}, \tag{21}$$

where k represents the k-th video segment.

- **Jain's Fairness Index**: The Jain's fairness index [7] defined as Eq.(22) 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:

$$J(y) = \frac{(\sum_{i=1}^{n} y_i)^2}{(n\sum_{i=1}^{n} y_i^2)}, \tag{22}$$

where y is a metric that needs to be measured for fairness, and n indicates that there are n users in total.

- **Aggregation Ratio**: NDN has ubiquitous caching and PIT that can achieve *Interest* aggregation. Then the aggregation among multiple clients can improve content distribution efficiency by reducing redundant data transmission. The larger the value of this metric, the higher the aggregation ratio requested by the user. The metric is defined as follows:

$$Ar(k) = \frac{agg(k)}{N}, \tag{23}$$

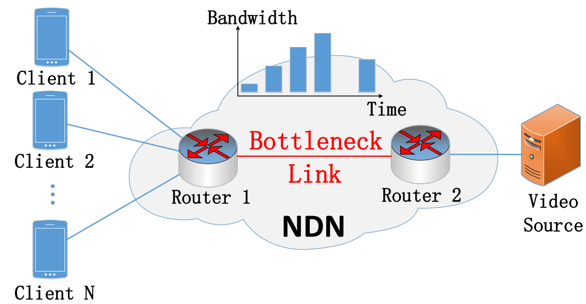


Fig. 5: Topology of simulation.

where $agg(k)$ represents the number of users for which $Interest$ aggregation has occurred among all N users requesting the k-th segment.

- **Amplification Factor of Bandwidth**: The bandwidth amplification factor is defined as the ratio of user's perceived bandwidth to the actually bandwidth used to transmit video data, where the user's perceived bandwidth is the sum of the bitrate of all video segments received by all users in the whole video playback process. This metric is used to measure the reduction of redundant data traffic as shown in Eq.(24):

$$Afb = \frac{\sum_{i=1}^{N}\sum_{k=1}^{K} br\_q_i(k)}{\sum_{k=1}^{K} C(k)}, \tag{24}$$

where $br\_q_i(k)$ denotes the bitrate of the k-th segment requested by the i-th user, and we have a total of N users and K segments. $C(k)$ indicates the bottleneck bandwidth spent to obtain the k-th segment.

B. Simulation Setup

We use ndnSIM as the simulation platform which is an open-source network environment simulator for NDN based on ns-3. The topology of the simulation and realistic experiments is shown in Fig.5, including multiple clients, routers, and a video source. The video source and clients are distributed at the edge of the network, and multiple clients will compete for bottleneck bandwidth resources, which is a typical scenario widely used in multi-client adaptive video stream research [14]-[16]. The links in the network are bidirectional, and the client-to-router bandwidth is set to 10Mbps, while the bottleneck link bandwidth between routers is set to 5Mbps. The bandwidth between the router and the video server is set to 5Mbps and 10Mbps in the scenario of two clients and five clients respectively.

The video source contains the video Big Buck Bunny [23] requested by the clients, and also contains the MPD file required for the requested video. The video file is divided into 299 fixed-length 2 seconds video segments, and each video segment is encoded into 20 different bitrate levels with a fixed resolution of 1080p [24] according to the MPEG-4 AVC standard. The average bitrate of each quality level is shown in Table I.

The cache size of each NDN node is 15MB, and the cache replacement strategy selects the Least Recently Used (LRU) algorithm, the routing strategy is selected as Best-Router. The

TABLE I: Requestable video bitrate

Index	Average Bitrate (bps)	Index	Average Bitrate (bps)
1	45652	11	791182
2	89283	12	1032682
3	131087	13	1244778
4	178351	14	1546902
5	221600	15	2133691
6	262537	16	2484135
7	334349	17	3078587
8	396126	18	3526922
9	522286	19	3840360
10	595491	20	4219897

simulation time is set to 200 seconds and 400 seconds in the scenario of two clients and five clients respectively. The client's local buffer size is set to 30 seconds, the screen resolution is 1920×1080, and start delay is set to 4 seconds. The client selects different algorithms to experiment in scenarios with fixed bottleneck bandwidth and scenarios with fluctuating bottleneck bandwidth, respectively. The setting of the four algorithms involved in the experiment are as follows:

- **RB**: The weighted average of predicted bandwidth and measured bandwidth is used as the predicted bandwidth to be used when making a bitrate decision for the next video segment with weight set to be 0.5.
- **BB**: The client reduces the bitrate of the next requested video segment when the local buffer size is less than 8s and increases it when the buffer size is larger than 14s. Otherwise, the requested bitrate remains unchanged.
- **BOLA**: The input weight parameter for prioritizing playback utility with the playback smoothness is set to be $\gamma = 2.5$.
- **GB**: In the scenario of two clients, the parameter $\alpha = 1.2$, $\beta = 2.5$, $\theta = 0.3$. And in the scenario of five clients, the parameter $\alpha = 0.5$, $\beta = 3.0$, $\theta = 0.3$.
- **Opt**: The algorithm is the optimal solution can be achieved in theory. Since Opt cannot be implemented on the realistic platform, the experimental results are only given in the simulation. In addition, the curves of aggregation ratio and the geometric mean of QoE are a straight line in all experiments, so they are omitted.

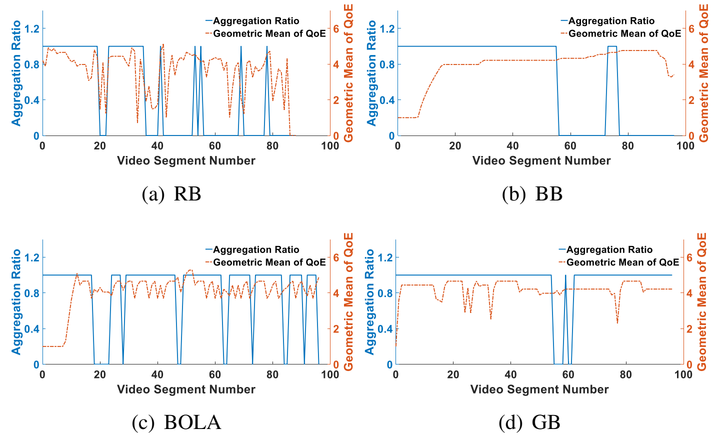
C. Simulation Results Analysis

*1) **Scenario with Fixed Bottleneck Bandwidth (Two Clients)**:* The final experimental results for scenario of two clients with fixed bottleneck bandwidth are shown in Fig.6, Fig.7 and Table II.

TABLE II: Components of QoE (two clients with fixed bottleneck bandwidth)

Algorithm	RB	BB	BOLA	GB	Opt
Video Quality	11.35	9.22	10.12	10.35	18
Video Quality Switching	0.14	0.04	0.17	0.09	0
Interruption Time	0.59	0	0	0	0

The RB algorithm makes bitrate decisions only based on estimated bandwidth, so it requests a higher bitrate of the video segments at the beginning. However, when two clients get video clips through the bottleneck link simultaneously, the



(a) RB (b) BB

(c) BOLA (d) GB

Fig. 6: Dynamic process of bitrate aggregation ratio and geometric mean of QoE during video playback (two clients with fixed bottleneck bandwidth).

available bandwidth may fluctuate. This will affect the accuracy of the bandwidth estimation, thus leading to inappropriate bitrate requests, resulting in frequent bitrate switching and playback interruption. Thus clients using the RB algorithm get the worst overall QoE.

The BB algorithm has an excellent performance in quality switching because it makes decisions only based on buffer size. However, the quality of video segments received by two clients is quite different, which results in unfairness. The average video quality of the BB algorithm is the lowest because the BB algorithm always makes bitrate decisions conservatively. As its goal is to maintain the fill ratio of the local buffer, the BB algorithm is relatively stable. Moreover, the BB algorithm only increases or decreases one quality level at a time, and does not change the quality of the requested video when the buffer is substantially unchanged, so it can achieve a fairly small value of average video quality switching.

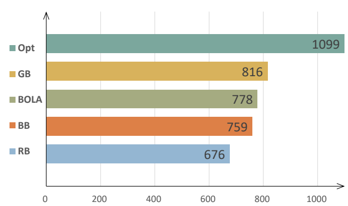
The BOLA algorithm has the largest quality switching because the BOLA algorithm increases or decreases the quality level by more than one level each time compared to the BB algorithm. Since the BOLA algorithm adopts the Lyapunov optimization theory to maximize video quality while minimizing playback interruptions, the bitrate decisions it makes are higher than the BB algorithm on average quality levels and it also does not suffer from playback interruptions.

The GB algorithm we proposed considers video quality switching and playback interruption as negative components of the payoff function, and the local buffer status is mapped as a corresponding parameter and introduced to it. Based on this payoff function, the client requests an appropriate bitrate of a video segment, which results in rare playback interrupting or drastic video quality switching during the experiment of the GB algorithm. More importantly, the GB algorithm achieves optimal overall QoE and fairness because of the largest aggregation ratio without the requirement of direct communication among the clients or scheduling by a central controller. This means the GB algorithm has the capability of proactive aggregation of bitrate requests among

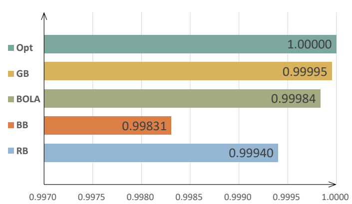
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3100768, IEEE Transactions on Multimedia

clients. The reason is that participants in the game are assumed to be perfectly rational, and the payoff function guides them to maximize their utilities when making decisions with the majority, which results in bitrate requests aggregation with high probability.

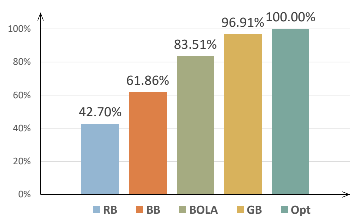
From the perspective of Jain's fairness index, the GB algorithm is still excellent. However, this fairness index is only explained from the perspective of overall quality. It can only explain the fairness of the algorithm to a certain extent. That is, although there may be only a small number of segments requesting the same bitrate for the two clients, after the end of the entire video playback, there may not be much difference in the sum of the quality levels of all the segments of the two clients. This can also explain why the performance of the RB algorithm is the worst, but it is the closest to the GB algorithm in the fairness index. Therefore, fairness should also be combined with the QoE geometric mean curve of Fig.12. From this point of view, the GB algorithm is still the best.



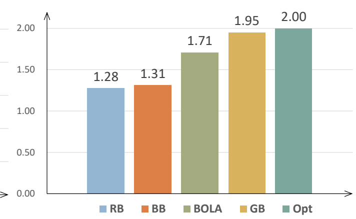
(a) Sum of all users'QoE



(b) Jain's fairness index



(c) Aggregation ratio



(d) Amplification factor of bandwidth

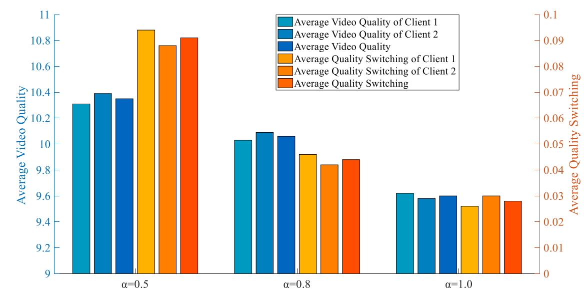
Fig. 7: Results of performance metrics for different algorithms (two clients with fixed bottleneck bandwidth).

For the purpose of showing the influence of the three weight parameters in our algorithm on the user's QoE, we conducted several comparative experiments (as shown in Fig.8) in this scenario. The value of α and β can reflect the preference of different users for videos. In order to test the impact of different preferences on the user's QoE, we set different α and β values in the experiment, and keep the value of θ constant. And θ is used to amplify punishment. The larger θ, the more conservative the user's strategy and the smaller the profit, but the more adaptable to scenarios where bandwidth changes drastically. Similarly, when conducting θ-related experiments, we keep α and β unchanged.

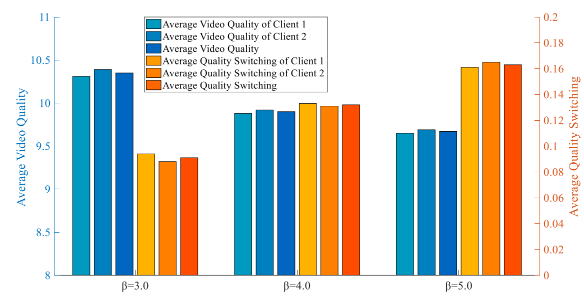
Fig.8 (a) verifies the impact of different values of α on the QoE. Since the interruption time is always zero, the interruption time is not shown in the figure (the same in (b) and (c)). The larger the α is, the more attention the user pays to quality switching, but the average video quality obtained by the user will also decrease accordingly.

Fig.8 (b) is the verification result of the influence of β. The larger β is, the more sensitive the user is to interruption of playback. From another perspective, the weight of video quality and quality switching in QoE has become lower. Therefore, the video quality will decrease and the quality switching will increase.

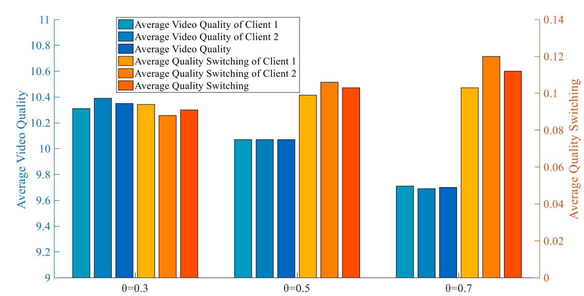
Fig.8 (c) verifies the impact of different values of θ on the QoE. The larger the θ, the more conservative strategy the user will adopt. That means that the bitrate of the video requested by the user will be lower, and the quality switching will not be greatly affected.



(a) The effect of α on video quality and quality switching (where β=3.0, θ=0.3)



(b) The effect of β on video quality and quality switching (where α=0.5, θ=0.3)



(c) The effect of θ on video quality and quality switching (where α=0.5, β=3.0)

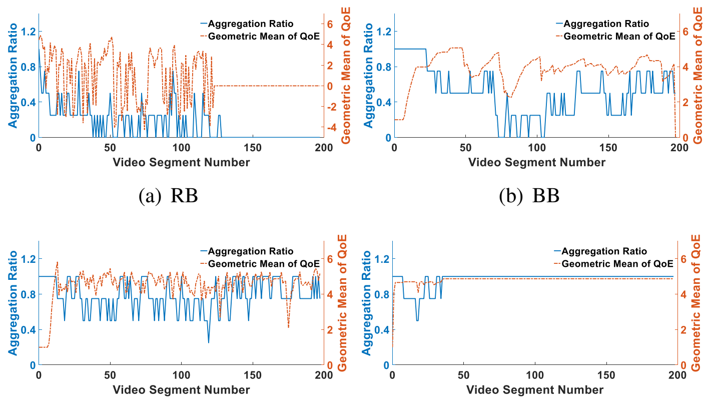
Fig. 8: The influence of different weight parameters on video quality and quality switching.

2) ***Scenario with Fixed Bottleneck Bandwidth (Five Clients):*** The final experimental results for the scenario of five clients with fixed bottleneck bandwidth are shown in Fig.9, Fig.10 and Table III.

TABLE III: Components of QoE (five clients with fixed bottleneck bandwidth)

Algorithm	RB	BB	BOLA	GB	Opt
Video Quality	10.84	9.18	12.11	12.77	18
Video Quality Switching	0.26	0.07	0.18	0.01	0
Interruption Time	2.62	0	0	0	0

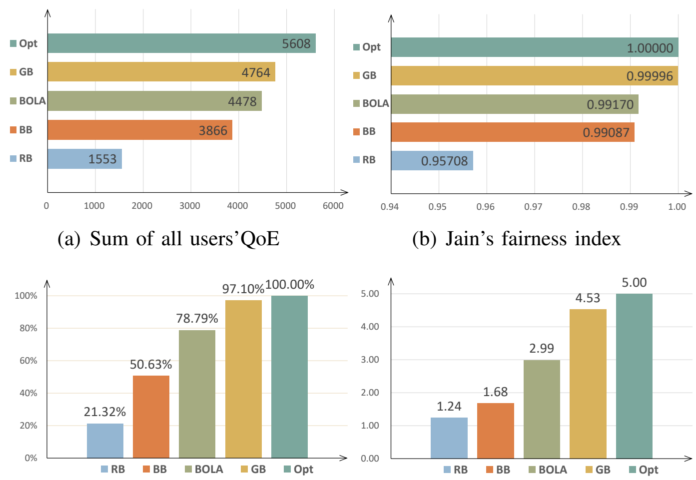
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3100768, IEEE Transactions on Multimedia



(a) RB (b) BB

(c) BOLA (d) GB

Fig. 9: Dynamic process of bitrate aggregation ratio and geometric mean of QoE during video playback (five clients with fixed bottleneck bandwidth).



(a) Sum of all users'QoE (b) Jain's fairness index

(c) Aggregation ratio (d) Amplification factor of bandwidth

Fig. 10: Results of performance metrics for different algorithms (five clients with fixed bottleneck bandwidth).

Compared with the scenario of two clients, the competition in the scenario of five clients becomes more complex. Even if the bandwidth of the bottleneck link is fixed, the accuracy of bandwidth prediction will be reduced due to incomplete state information of the client-side and more combinations of video request aggregation. Therefore, in this case, the performance of the RB algorithm is relatively worse. It is worth noting that the QoE geometric mean of the RB algorithm has a different coordinate range than the other three. This is because the playback interruption of the RB algorithm is too frequent and the interruption time is too long so that the QoE of a single video segment obtained by the QoE calculation Eq.(1) will appear to be less than zero (an extremely poor experience).

The BB and BOLA algorithms take into account the local cache on the client, reducing the dependence on the accuracy of the bandwidth prediction, and thus can obtain better performance than the RB algorithm.

As to the GB algorithm, due to the formulation of the payoff function we designed, the optimal strategy to maximize the payoff for each client is bitrate requests aggregation. This scheme can cope with more complex situations, so a better aggregation ratio and amplification factor of bandwidth are obtained compared to the two client scenarios.

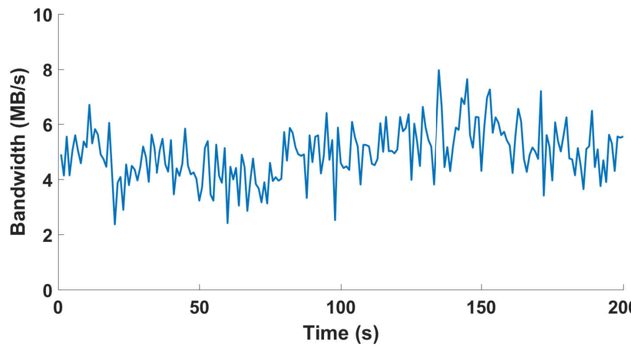


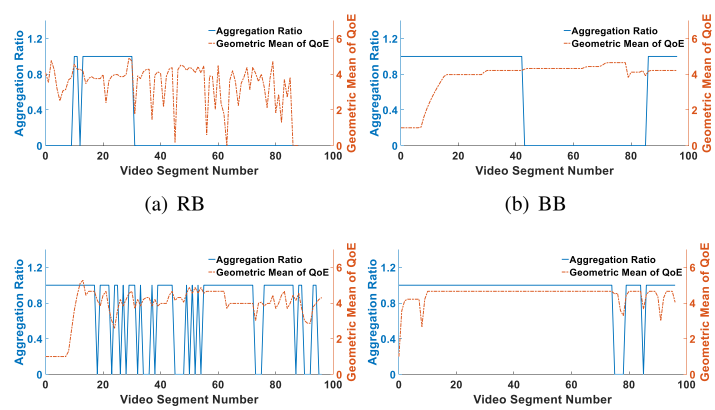
Fig. 11: Bandwidth trace.

*3) **Scenario with Fluctuating Bottleneck Bandwidth (Two Clients)**:* In this scenario, the bandwidth of the bottleneck link changes according to a preset trajectory as Fig.11. The experimental results of this scenario are shown in Fig.12, Fig.13 and Table IV.

TABLE IV: Components of QoE (two clients with fluctuating bottleneck bandwidth)

Algorithm	RB	BB	BOLA	GB	Opt
Video Quality	10.67	9.03	9.36	11.50	18
Video Quality Switching	0.14	0.03	0.15	0.06	0
Interruption Time	0.56	0	0	0	0

Compared with the fixed bandwidth scenario, the performance of all algorithms in the scenario with fluctuating bottleneck bandwidth has decreased to different degrees. The reason is that the RB, BOLA, and GB algorithms all use the estimated bandwidth more or less. Although the BB algorithm makes bitrate requests decisions based on the locally cached video segments, the transmission of video segments is also affected by fluctuating bandwidth.

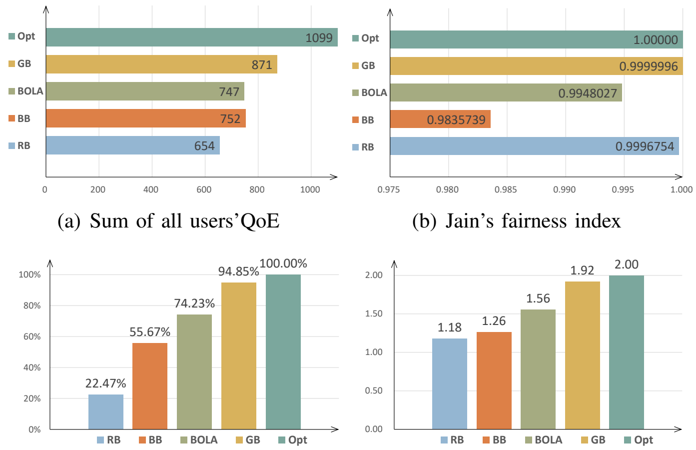


(a) RB (b) BB

(c) BOLA (d) GB

Fig. 12: Dynamic process of bitrate aggregation ratio and geometric mean of QoE during video playback (two clients with fluctuating bottleneck bandwidth).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3100768, IEEE Transactions on Multimedia



(a) Sum of all users'QoE

(b) Jain's fairness index

(c) Aggregation ratio

(d) Amplification factor of bandwidth

Fig. 13: Results of performance metrics for different algorithms (two clients with fluctuating bottleneck bandwidth).

Thanks to its powerful adaptive capabilities of the GB algorithm, the aggregation ratio of it is still much larger than the other three algorithms. The reason is that each user takes into account the optimal decisions that other users can make, and then chooses the video bitrate that maximizes their revenue without interrupting playback.

D. Summary

In summary, for the case of fixed bottleneck bandwidth, we performed two client and five client simulation experiments. With the number of clients increasing from 2 to 5, the results show that the performance of our algorithm is equally robust and even better, while the performance of other algorithms has dropped a lot. This is mainly due to our proactively considering the aggregation of multiple client requests to the greatest extent possible. The greater the number of clients, the greater the likelihood of aggregation, and the better the experimental results will be.

In the case of fluctuating bottleneck bandwidth, we made the number of clients keep consistent and then verified the performance of our algorithm. Under the premise that the bandwidth changes from constant to additional fluctuation, although the performance of all algorithms is degraded, the performance degradation of our algorithm is the smallest. With its powerful adaptive features, the GB algorithm allows each user to consider the optimal decisions others can make, and then choose the video bitrate that maximizes his revenue without interrupting playback. For a single user, after several segments have been tried, they can achieve bitrate aggregation with other users, so that their respective gains can be effectively guaranteed.

VI. Performance Evaluation for Realistically Networking Scenarios

In order to further verify the performance of the GB algorithm and exhibit its feasibility and applicability, we develop a realistically experimental platform called DAS-NDN and perform algorithm performance evaluation on it.

A. DAS-NDN implementation and Experiment Environment Construction

We develop the DAS-NDN based on Libdash ([21], [22]) which is the standard for adaptive video stream implementation in HTTP with a good adaptive algorithm interface. The DAS-NDN consists of the following four modules:

- **Video Playback Module**: The video playback module is the interface for the user to watch the video by decoding the video segments and interacting with the video service. This module is developed using QT tools.
- **Status Collection Module**: This module is responsible for collecting the related statuses information, such as the local buffer size of the client, network bandwidth, delay, etc. These statuses are important references for decision-making by the bitrate adaptation module.
- **Adaptive Bitrate Decision Module**: The adaptive bitrate decision module invokes the corresponding algorithm according to the user's choice, and performs bitrate decision, and then sends the decision result to the NDN transmission module. So far, we have implemented RB, BB, BOLA, and GB algorithms in this module.
- **NDN Transmission Module**: This module is an interface between the NDN network and the adaptive video stream clients. It is responsible for wrapping the requested information into the form of *Interest* packets in NDN and receiving the NDN formatted data packets returned by the video source or router.

Due to limited laboratory conditions, we only set up a relatively simple scenario shown in Fig.5. Although such a topology is simple but typical, it is widely used in the evaluation of adaptive video schemes. The video for testing is *Big Buck Bunny*, an animated type of video.

All video segments and MPD files are available in the video source. The video segments are encoded according to the AVC standard and available for request. When the video source receives the *Interest* packets, it parses it and, correspondingly, returns the packet encapsulated according to the NDN protocol. To evaluate the performance of the algorithms, the bandwidth limit module uses the Linux Traffic Control tool (TC tool) to limit the bandwidth so that the bottleneck bandwidth between the two routers fluctuates between 4Mbps and 7Mbps.

Two clients are watching the video at the same time, and the duration of playback is the entire video. Clients to Router 1 and Router 2 to the video source are directly connected through the network cable with 100Mbps bandwidth, and the bandwidth between Router 1 and Router 2 is the bottleneck link. Each client is equipped with an i7-3770 processor and 16GB memory, while ndn-cxx (NDN C++ library with eXperimental eXtensions) 0.5.0 and NFD 0.5.0 are installed running on 64-bit Ubuntu 16.04 LTS.

B. Experiment Results Analysis

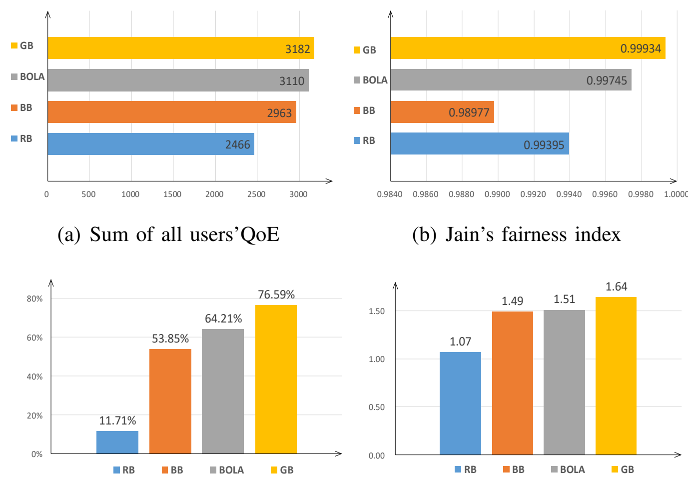
The DAS-NDN we developed can run stably on a realistic NDN platform. We evaluate RB, BB, BOLA, and GB algorithms using DAS-NDN, and the experimental results are shown in Fig.14 and Table.V. From the perspective of the sum

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3100768, IEEE Transactions on Multimedia

of all users' QoE, the evaluation result of the GB algorithm in the realistic platform is still the best.

TABLE V: Components of QoE (realistically modeled networking scenario)

Algorithm	RB	BB	BOLA	GB
Average Video Quality	12.83	14.37	15.20	15.38
Average Video Quality Switching	0.32	0.04	0.03	0.04
Average Interruption Time	0.41	0	0	0



(a) Sum of all users'QoE (b) Jain's fairness index

(c) Aggregation ratio (d) Amplification factor of bandwidth

Fig. 14: Results of performance metrics for different algorithms (realistically modeled networking scenario with two clients).

In a realistic network environment, there exists much more unknown and uncontrollable factors, which will have a great impact on the performance of the algorithms, so the aggregation ratio of GB algorithm is a bit lower than that in the simulation scenario, but it is still much higher than the other three algorithms.

The main reason is that the playback time in the realistic platform is about three times longer than in the simulation environment (more segments are counted). Because we take into account the complexity of the realistic network environment, the use of relatively long playback times allows for more accurate experimental results.

Whereas the average video quality is higher and the average interruption time is smaller in the actual platform experimental results because the bottleneck bandwidth is greater in the realistic environment than in the simulation environment. We can find that the average video quality of the RB algorithm is the highest in the simulation experiment, but becomes the lowest in the actual experiment. Because all algorithms have the same simulation duration in simulation experiments, clients using RB algorithms always request too high a bitrate resulting in more playback interruptions due to incorrect estimates of bandwidth. However, the corresponding number of video segments obtained by RB is also much lower than other algorithms, resulting in the highest average video quality. But in the actual platform experiments, the client ended up playing the whole video (with the same number of segments acquired) regardless of which algorithm was used, and RB certainly had the lowest average video quality as we expected.

In terms of fairness, aggregation radio, and bandwidth resource utilization, we can also draw similar conclusions to simulation experiments, although the relevant performance metrics are declined due to the complexity of realistic network environments. We could find that the aggregation ratio of the GB algorithm is much higher than that of other algorithms, which is the fundamental reason why our algorithm performs well. We model the adaptive video streaming system for multiple clients over NDN as a Bayesian game. Based on the pre-designed payoff function, the client can make the bitrate decision according to the solution of Bayesian Nash Equilibrium. In the process of repeated games, a more accurate distribution of *type* can be obtained. Thus, video segments requested by different clients are much more likely to be aggregated. Accordingly, we demonstrate the effectiveness of the proposed algorithm in a realistic network scenario.

VII. Conclusion

In this paper, we propose a multi-client dynamic adaptive video streaming algorithm named GB. We introduce game theory into adaptive video streaming for multiple clients scenarios of NDN and model it as a Bayesian game, and the clients can make effective bitrate decisions according to Bayesian Nash Equilibrium of this game. The proposed GB algorithm for NDN can achieve proactive aggregation of requests among clients as much as possible without client-to-client direct communication or scheduling by a central controller. We have performed detailed evaluations and contrast experiments for the proposed algorithm with others on the simulation platform and the realistic network environment. Experimental results show that, compared with the other three comparative algorithms, the GB algorithm can produce a higher performance in overall QoE, fairness, and bandwidth resource utilization, etc. It is worth mentioning that our algorithm needn't change the architecture of the current NDN network and add no explicit signaling between clients. However, because the payoff function pair involves too many related parameters, the function we currently proposed is only designed according to our experience, and we are sure that it still has a lot of space for improvement. A possible solution is introducing a neural network instead of a hand-designed payoff function pair that can guide the clients to make better bitrate decisions.

Acknowledgements

This work is supported by the National Key R&D Program of China under grant No. 2020YFA0711400, the National Science Foundation of China under grant No. 61673360, and the CETC Joint Advanced Research Foundation under grant No. 6141B08080101.

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] J. Samain et al., "Dynamic Adaptive Video Streaming: Towards a Systematic Comparison of ICN and TCP/IP," *IEEE Trans. Multimed.*, vol. 19, no. 10, pp. 2166-2181, Oct. 2017.

[4] NDN Team. *NDN Forwarding Daemon*, accessed on 2017. [Online]. Available: http://named-data.net/doc/NFD

[5] J. Thijssen, *Noncooperative Game Theory*, York, 2015.

[6] R. Milner, *A Calculus of Communicating System*, Springer Berlin Heidelber, 1980.

[7] R. Jain, D. Chiu, W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," *Tech. Rep. TR-301*, 1984.

[8] T. Y. Huang, R. Johari, N. Mckeown, et al., "A buffer-based approach to rate adaptation: evidence from a large video streaming service," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 44, no. 4, pp. 187-198, 2014.

[9] B. Rainer, D. Posch, H. Hellwagner, "Investigating the performance of pull-based dynamic adaptive streaming in ndn," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2130-2140, 2016.

[10] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal Bitrate Adaptation for Online Videos," in *Proc. IEEE INFOCOM*, 2016.

[11] S. Mastorakis, A. Afanasyev, I. Moiseenko, et al., "ndnSIM 2.0: A new version of the NDN simulator for NS-3," *Tech. Rep. NDN-0028*, 2015.

[12] W. Liu, G. Zhang, Q. Gao, "SVC-aware Cache Coordination Schemes for Dynamic Adaptive Streaming in Named Data Networking," in *Proc. IEEE ICC*, 2020, pp. 1-6.

[13] M. Hassan, H. Salat, M. Conti, et al., "CoMon-DAS: A Framework for Efficient and Robust Dynamic Adaptive Streaming over NDN," in *Proc. IEEE ISCC*, 2019, pp. 1-7.

[14] S. Awiphan, K. Poobai, J. Katto, "Proxy-Assisted Rate Adaptation for 4K Video Streaming on Named Data Networking," in *Proc. IEEE GCCE*, 2019, pp. 181-183.

[15] R. Li, K. Matsuzono, et al., "Consecutive Caching and Adaptive Retrieval for In-Network Big Data Sharing," in *Proc. IEEE ICC*, 2018, pp. 1-6.

[16] C. Kreuzberger, B. Rainer, H. Hellwagner, "Modelling the impact of caching and popularity on concurrent adaptive multimedia streams in information-centric networks," in *Proc. IEEE Int. Conf. Multimed. Expo Workshops*, 2015, pp. 1-6.

[17] B. J. Villa, P. E. Heegaard, A. Instefjord, "Improving fairness for adaptive http video streaming," in *Meeting Eur. Network Universities Companies Inf. Commun. Eng.*, 2012, pp. 183-193.

[18] J. Jiang, V. Sekar , H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 326-340, 2014.

[19] Z. Li, X. Zhu, J. Gahm, et al., "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE J. Sel. Areas Commun.*, vol.32, no. 4, pp. 719-733, 2014.

[20] R. Li, H. Asaeda, "MWBS: An Efficient Many-to-Many Wireless Big Data Delivery Scheme," *IEEE Trans. Big Data*, vol. 6, no. 4, pp. 233-247, 2020.

[21] C. Mueller, S. Lederer, J. Poecher, and C. Timmerer, "Demo paper: Libdash-An open source software library for the MPEG-DASH standard," in *Proc. IEEE Int. Conf. Multimed. Expo Workshops*, 2013, pp. 1-2.

[22] P. Georgopoulos, Y. Elkhatib, M. Broadbent, et al., "Towards network-wide QoE fairness using openflow-assisted adaptive video streaming," in *Proc. ACM SIGCOMM Workshop Future Human-Centric Multimed. Netw.*, 2013, pp. 15-20.

[23] S. Lederer, C. Timmerer, "Dynamic adaptive streaming over HTTP dataset" in *Proc. ACM Conf. Multimed. Syst.*, 2012, pp. 89-94.

[24] DASH Dataset at ITEC/Alpen-Adria-Universitat Klagenfurt. ITEC - Dynamic Adaptive Streaming over HTTP, accessed on 2018. [Online]. Available: https://dash.itec.aau.at/dash-dataset

[25] H. Yuan, X. Wei, F. Yang, J. Xiao and S. Kwong, "Cooperative bargaining game-based multiuser bandwidth allocation for dynamic adaptive streaming over HTTP," *IEEE Trans. Multimed.*, vol. 20, no. 1, pp. 183-197, 2018.

[26] Z. Liu and Y. Wei, "Hop-by-hop Adaptive Video Streaming in Content Centric Network," in *Proc. IEEE Int. Conf. Commun.*, May. 2016, pp. 1-7.

[27] 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.

[28] B. Alt, T. Ballard, R. Steinmetz, A. Rizk, and H. Koeppl, "CBA: Contextual Quality Adaptation for Adaptive Bitrate Video Streaming," in *Proc. IEEE INFOCOM Int. Conf. Computer Commun.*, , 2019.

[29] 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.

[30] Z. Akhtar, YS. Nam, R. Govindan, et al., "Oboe: auto-tuning video ABR algorithms to network conditions," in *Proc. ACM SIGCOMM*, 2018.

[31] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in *Proc. ACM SIGCOMM*, 2017.

[32] H. Yuan, H. Fu, J. Liu, et al., "Non-Cooperative Game Theory Based Rate Adaptation for Dynamic Video Streaming over HTTP," *IEEE. Trans. Mob. Comput.*, vol. 17, no. 10, pp. 2334-2348, 2018.

[33] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *Proc. ACM SIGCOMM*, 2015.

[34] 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.

[35] Z. Xu, X. Zhang and Z. Guo, "QoE-Driven Adaptive K-Push for HTTP/2 Live Streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no.6, pp. 1781-1794, 2018.

[36] L. Zou, R. Trestian, GM. Muntean, "E3DOAS: Balancing QoE and energy-saving for multi-device adaptation in future mobile wireless video delivery," *IEEE Trans. Broadcast.*, vol. 64, no. 1, pp. 26-40, 2017.

[37] A. Bentaleb, AC. Begen, S. Harous, et al., "Want to play dash ?: A game theoretic approach for adaptive streaming over http," *ACM Multimed. Syst. Conf.*, 2018.

[38] X. Yin, M. Bartulovi, V. Sekar and B. Sinopoli, "On the efficiency and fairness of multiplayer HTTP-based adaptive video streaming," in *Proc. Am. Control Conf.*, 2017.

[39] H. Hu, et al., "An epistemic analysis of the Harsanyi transformation," *Int. J. Game Theory*, vol. 30, no. 4, pp. 517-525, 2002.

[40] J. Monks, A. Olaru, G. Muntean, "Buffer-Aware Dynamic Adaptive Streaming over Content Centric Networks," in *Proc. IEEE BMSB*, 2019, pp. 1-6.