Decentralized Optimization for Multicast Adaptive Video Streaming in Edge Cache-Assisted Networks

Lujie Zhong, Mu Wang, *Member, IEEE*, Changqiao Xu, *Senior Member, IEEE*, Shujie Yang, and Gabriel-Miro Muntean, *Fellow, IEEE*

***Abstract*—Adaptive streaming based on DASH offers personalized video experience and smooth playback by allowing dynamical adjustments of the video bitrate to the variations of network conditions. This is especially important for current and future Internet video streaming applications, including emerging ones such as virtual reality-based, as adaptive streaming plays a key role in providing high quality viewing experience, especially in limited bandwidth delivery environments. To enable this promising avenue in a 5G context, efforts are made to consider it alongside multicast and edge caching, as part of the next generation communication technology. In this paper, we model the adaptive streaming transmission problem in a mobile scenario as a multi-source multicast multi-rate problem (MMMP) whose linear relaxation is concave. We decompose the problem in terms of clients and propose the distributed delivery algorithm (DDA). The computation complexity, convergence and time-varying adaptation of DDA are theoretically analyzed. Additionally, to further reduce the computation complexity of the solution, a heuristic approximation method (H-DDA) based on the physical meaning of the problem is proposed and it is also shown how H-DDA converges to the optimal value by numerical means. Finally, we conduct a series of simulation tests to demonstrate the superiority of the proposed HDDA in comparison with other state-of-art solutions.**

***Index Terms*—Adaptive streaming, multicast delivery, rate control, dual optimization theory.**

I. Introduction

ADAPTIVE streaming such as dynamic video streaming over HTTP (DASH) [1] enable video delivery based on diverse representations and dynamic content adjustment to match network bandwidth variations and different user equipment characteristics. The latest increases in the demand for bandwidth and expectations in terms of viewing quality make adaptive streaming necessary for various emerging video applications. For example, virtual reality (VR) [2], which is a type of omnidirectional video with ultra large bandwidth requirements, heavily relies on the adaptive streaming technology to deliver the high definition content within the viewer's field-of-view (FoV) only. By not delivering the whole image, any unnecessary bandwidth consumption caused by the potential delivery of the rest of the image is avoided. Most of current DASH-based solutions have been proposed for conventional wired networks [3] and broadband wireless networks [4], [5]. By introducing ubiquitous edge caching and multicast support for the wireless connections, current cache-assisted mobile networks enable large scale low latency video services. Given the fact that caching video content at the edge not only reduces the delivery latency, but also simplifies the multicast design thanks to the multicast feature of wireless communications, integrating edge caching into the video system facilitates multicast video delivery. With such advantages, there is a natural interest in proposing solutions for adaptive video streaming in such an environment [6], [7]. Yet, achieving optimal delivery over the cache-assisted mobile networks is non-trivial.

On one hand, adaptive streaming refers to dynamical selection of the video bitrate in order to both optimize user quality-of-experience (QoE) and maximize the utilization of bandwidth resources while avoiding network congestion [2], [13], [14]. Adaptive streaming schemes require to frequently rearrange the bitrate selection policy to adapt the randomness of the wireless communications, user preferences, etc. On the other hand, despite the benefits brought by multicast and ubiquitous caching [17], these features also make traditional rate adaption methods [8], [9] impossible to use. An important issue is that the fully distributed aspect of ubiquitous caching triggers the requirement of optimizing overall user bitrate by using a decentralized method based on local information. In addition, conventional adaptive streaming serves video clients separately and adjusts user streaming rate individually [1], [14]. However, this one-to-one design paradigm is not appropriate in the context of one-to-many multicast delivery.

In this paper, we focus on proposing a decentralized method for adaptive multicast video streaming in a cache-assisted mobile network environment. First, we mathematically model the adaptive streaming problem and then propose an optimal decentralized delivery algorithm (DDA) which enables each video client optimize its bitrate without coordinating with other clients. Moreover, we further propose a heuristics rate adaption algorithm that significantly reduces the computation load while approximating the optimal value derived by DDA.

Manuscript received 13 December 2022; revised 24 February 2023; accepted 28 February 2023. Date of publication 24 March 2023; date of current version 6 September 2023. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62101301, Grant 62225105, and Grant 61872253; in part by the Science Foundation Ireland (SFI) under Grant 21/FFP-P/10244 (FRADIS) and Grant 12/RC/2289_P2 (INSIGHT); and in part by the China Postdoctoral Science Foundation under Grant 2021M691787. *(Corresponding authors: Gabriel-Miro Muntean; Mu Wang.)*

Lujie Zhong is with the Information Engineering College, Capital Normal University, Beijing 100048, China (e-mail: zhonglj@cnu.edu.cn).

Mu Wang, Changqiao Xu, and Shujie Yang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: muwang@bupt.edu.cn; cqxu@bupt.edu.cn; sjyang@bupt.edu.cn).

Gabriel-Miro Muntean is with the School of Electronic Engineering, Dublin City University, Dublin 9, D09 DXA0 Ireland (e-mail: gabriel.muntean@dcu.ie).

Digital Object Identifier 10.1109/TBC.2023.3254165

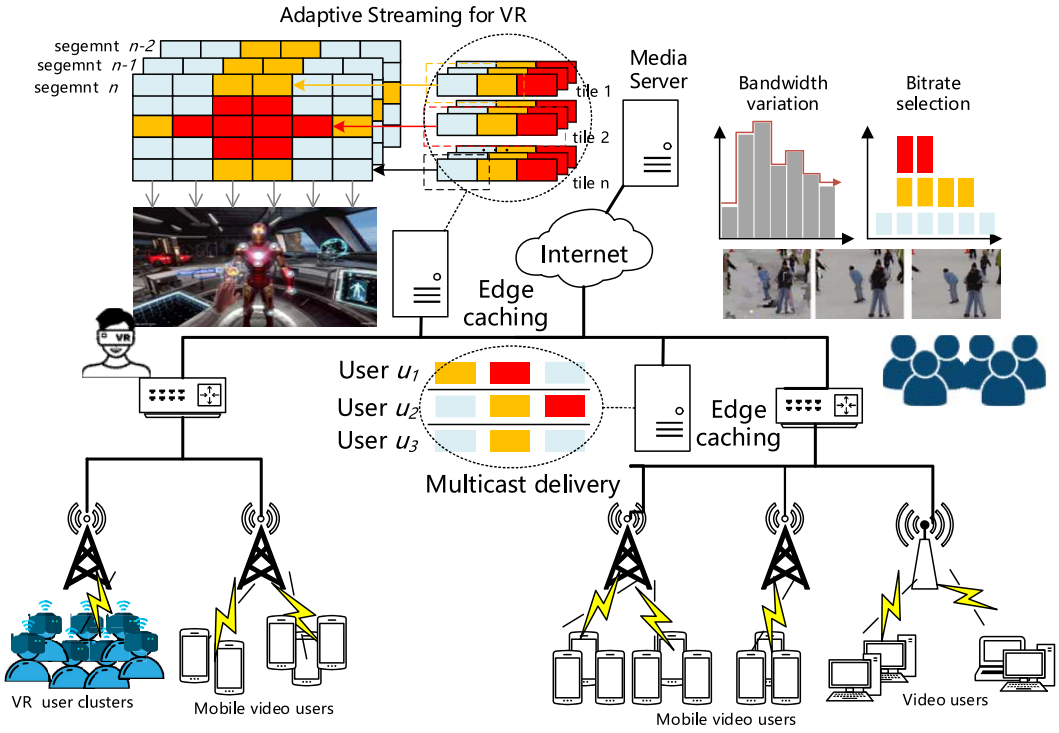


Fig. 1. An illustration of cache-assisted mobile adaptive streaming.

We present a series of simulation tests which demonstrate the close to optimal performance of the proposed algorithm, and show how our algorithm outperforms other state-of-art solutions. The main contributions of this paper are:

1) *Multisource Multicast Rate-Adaptive Problem:* We formulate mathematically the optimal adaptive video streaming in cache-assisted mobile networks as a multi-source multicast multi-rate problem (MMMP). We then introduce a linear relaxation of MMMP and prove its concavity, demonstrating that it has a unique optimal solution.
2) *Distributed Delivery Algorithm:* We further decompose MMMP in terms of the end users and prove the equivalence between the original MMMP and the decomposed problems. Furthermore we propose DDA, a decentralized algorithm which achieves optimal rate adaptation. We also extend our algorithm to the omnidirectional video applications such as VR.
3) *Heuristic Distributed Delivery Algorithm:* By observing the physical meaning of the problem, we further propose a heuristic rate adaption algorithm (H-DDA) that achieves a similar performance, but yet dramatically reduces the computation load in comparison with the original DDA.

Our algorithms are implemented and involved in simulations, whose results show how they approximate the theoretical optimal and outperform state-of-art solutions.

II. Background and Related Works

A. Caching-Assisted Mobile Adaptive Streaming Background

Fig. 1 illustrates the process of provisioning adaptive video streaming services in cache-assisted mobile scenarios. Video clients can retrieve the content from a nearby edge caching node instead of the far-end media server. Each client uses the adaptive streaming control module to determine the appropriate representation of the desired video based on their network conditions. This is especially important in VR applications [23], where the image is tiled and each tile has multiple representations. The adaptive streaming control module selects the bitrates for each tile according to the network conditions and user viewport as well. Adaptive streaming supports the video provider to deliver only the users' area of interest rather than the whole image at high quality, saving bandwidth without impairing the viewing experience.

B. Related Works

Numerous studies have focused on improving the performance of adaptive video streaming. For example, Yuan et al. in [15] proposed an ensemble rate adaptation framework which aims to take advantage of the benefits of multiple rate adaptation methods. The proposed framework mainly consists of two modules, a method pool to store the rate adaptation policy and a method controller to decide the policy to use. By constructing a two layer network structure, a distributed joint optimization algorithm for adaptive video streaming which aims to maximize the total user demand rate is proposed in [16]. Furthermore, a modified algorithm with a practical caching strategy is also designed in order to support realistic implementation. Several recent studies attempts to apply the machine learning method to deal with the high dynamic network conditions when adaptively streaming the video content. For example, in [18], a Q-learning model is applied to generate adaptive streaming schemes for 5G multimedia services with the aim

to preserve both energy efficiency and user QoE. TCLiVi in [19] applies the deep reinforcement learning to control the bitrate selection for adaptive streaming. A major difference between our work and current reinforcement learning-based studies is that we take the multicast into account instead of only considering the case of end-to-end video delivery. Besides, reinforcement learning requires pre-training of a learning model which can be time consuming and requires a-priori knowledge of the network, which can be practically difficult.

Most studies on cache-assisted adaptive streaming focus on cache placement and are not focused on in this work. Zhang et al. [17] proposed VISCA, which integrates the edge caching capacity to enhance the streaming performance. Moreover, a novel Adaptive BitRate (ABR) algorithm decides the bitrate and video chunk source by considering network conditions, QoE objectives, and edge resource availability jointly is then proposed. VISCA also uses the super resolution method to enhance the low-quality data. Liu and Wei [20] designed a hop-by-hop adaptive streaming control, which sets a scheduling window at each switch to limit the data transmission rate according to the one-hop link capacity. Furthermore, a priority-based data delivery scheme is proposed to enable popular and lowest representation video content to be delivered preferentially. However, this method only adapts video rate at each node individually, and it is difficult to achieve overall clients bitrate adaption optimization without coordinating with each other. Eswara et al. [21] formulated the resource allocation problem for adaptive streaming as a stochastic optimization problem with the purpose to optimize the long term QoE metrics. However, the formulated problem treats each flow individually and ignores the multicast feature of the wireless communication, decreasing the transmission performance.

In the context of the above discussion, a distributed method that not only supports multicast, but also provides optimal rate adaptation is required for cache-assisted scenarios and is proposed in this paper.

III. System Model and Problem Formalization

A. System Model

Table I shows the notations used in this paper. We consider a network of N nodes including cache carriers, switches content sources and users that communicate with each other over a given connected, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$. $\mathcal{V}$ and $\mathcal{L} \subseteq \mathcal{V} \times \mathcal{V}$ denote the set of nodes and network links, respectively. Let $\mathcal{S} = \{1, \ldots, S\} \in \mathcal{V}$ be the set of video providers in the network and $\mathcal{U} = \{1, \ldots, U\} \in \mathcal{V}$ the set of end users. We define the path from client i to its video provider j as p_j, i.e., $p_j \triangleq \{l_{i,s_1}, l_{s_1,s_2}, l_{s_2,s_3}, \ldots, l_{s_n,j}\}$, where $l_{x,y}$ indicates the link between nodes x and y. Due to the in-network caching, intermediate nodes can also be treated as providers, namely, for all $k = 1, 2, 3, n$, $s_k \in \mathcal{S}$.

In our adaptive model, scalable video coding (SVC) [22] is used to encode video content into a base layer and several enhancement layers. Video clients can either decode the video with only the base layer or with base plus multiple enhancement layers. The more enhancement layers decoded, the better quality of video can be presented. Let videos in $\mathcal{G}$ consist of m enhancement layers, and let b_1 and h_k be the bitrate of base layer and each enhancement layer k, respectively. Accordingly, possible requested video bitrates are $\mathbb{B} = (b_1, b_1 + h_1, b_1 + h_1 + h_2, b_1 + h_1 + \cdots + h_m)$. Denote $b_{\min} = b_1$ and $b_{\max} = b_1 + h_1 + \cdots + h_m$. With the layered coding property of SVC, content providers can serve multiple request of the same video with different bitrates by multicasting the highest requested bitrate, and the switch forwards the layers of data to clients according to the request of client.

TABLE I
Notifications Used in Problem Formulation

Notation	Definition
N	the number of nodes in the network
$\mathcal{V}, \mathcal{L}$	set of the nodes and links
$\mathcal{S}$	the set of video providers
$\mathcal{U}$	the set of end users
$l_{x,y}$	link between x and y
$\mathcal{G}$	the set of videos
$\mathbf{x}$	the vector denotes the selected bitrate of all clients
$x_{i,j}$	transmission rate between i and j
c_l	link capacity of l
$J(x)$	QoE value when the bitrate is x
$s_i(u)$	client set using provider i
$s_i(u)_l$	client set using provider i via link l
$l(s)$	set of providers using link l
θ	a nonegative weight within $[0, 1]$
x_k^l	the maximum bitrate of users using link l
${x_{i,j}}^*$	the optimal value of transmission rate between i and j
$b_{\max}(b_{\min})$	the bandwidth required by highest (lowest) representation
$\boldsymbol{\lambda}_{\boldsymbol{p}}^*, \boldsymbol{\upsilon}^*$	Lagrange operators

In our solution, we attempt to optimize the rate adaptation and maximize user QoE. As we select the bitrate to optimise both network bandwidth utilisation and user QoE, we use the QoE model proposed in [25] and introduced next[1]:

$$J(x) = 4.75 - 4.5e^{-0.77x}. \tag{1}$$

B. Problem Formalization

The objective of the rate adaption algorithm is to chose a rate adaption strategy $\boldsymbol{x}$ to maximize the overall user QoE given the network capacity constraints. Let strategy vector $\boldsymbol{x} = \{x_{1,1}, \ldots, x_{i,j}, \ldots, x_{S,U}\}$, where $x_{i,j}$ implies the delivery rate of client j receiving video from i. We represent the capacity of links in $\mathcal{L}$ as a vector $\boldsymbol{c} = \{c_1, c_2, c_3, \ldots, c_L\}$. The objective function $f(\boldsymbol{x})$ is defined as the overall sum of client QoE, i.e., $f(\boldsymbol{x}) = \sum_{i \in \mathcal{S}} \sum_{j \in s_i(u)} J(x_{i,j})$, where $J(\cdot)$ is given in eq. (1). Considering the above objective function, the rate adaptation streaming problem can be referred to as the following multicast multi-sources multi-rate problem (MMMP): **P1**.

P1:

$$\max \quad \sum_{i \in \mathcal{S}} \sum_{j \in s_i(u)} J\left(x_{i,j}\right) \tag{2}$$

$$s.t \quad \sum_{i \in l(s)} \max_{j \in s_i(u)_l} x_{i,j} \leq c_l \quad l \in \mathcal{L} \tag{3}$$

$$x_{i,j} \in \mathbb{B} \quad i \in \mathcal{S}, j \in s_i(u) \tag{4}$$

[1]Note that other utility functions with properties of concavity and twice differential bitrate can also be employed into our problem.

where $s_i(u)$ is the clients set of providers i, $l(s)$ denotes the set of providers that use link l, $s_i(u)_l$ denotes the set of clients using link l to access videos from i. Accordingly, $\max_{j\in s_i(u)_l} x_{i,j}$ indicates the maximum bitrate over link l of users in the multicast tree rooted at i; we define this bitrate as the *provider rate* of i over l. The constraints from eq. (3) ensure that in a multicast scenario, for any link l, the total sum of *provider rates* cannot exceed the capacity c_l. Constraints from eq. (4) indicate that each client selects a bitrate from $\mathbb{B}$ to request. If $J(x_{i,j})$ is given by eq. (1), (2) and (3) are concave and convex [26], respectively. However, $\mathbb{B}$ is a discrete set, making the **P1** hard to be solved. Instead, we consider the linear relaxation of the MMMP as follows:

P2:

$$\max_{\boldsymbol{x}\in[b_{\min},b_{\max}]^U} \quad \sum_{i\in\mathcal{S}}\sum_{j\in s(u)} J\left(x_{s,j}\right) \tag{5}$$

$$s.t \quad \sum_{i\in l(s)} \max_{j\in s_i(u)_l} x_{i,j} \le c_l \quad l\in\mathcal{L} \tag{6}$$

where $\boldsymbol{x}\in[b_{\min},b_{\max}]^U$ indicates that the rate adaption strategy can be chosen from a continuous U-dimensional close space, which is considered as the relaxation of constraint from eq. (4) in **P1**. The closure space $[b_{\min},b_{\max}]^U$ is a convex set because $\forall \boldsymbol{x},\boldsymbol{y}\in[b_{\min},b_{\max}]^U$ and $0<\theta<1$, we have $\theta\boldsymbol{x}+(1-\theta)\boldsymbol{y}\in[b_{\min},b_{\max}]^U$. Therefore, problem **P2** is a concave optimization [26] whose maximum value is unique.

In spite of adopting multi-sources and multicast features, **P2** can be easily generalized to other scenarios with minor modifications. For example, to apply **P2** in a scenario with a single provider concurrently delivering multiple videos, we can split the provider i with n video flows into n virtual source nodes. Virtual node i_k corresponding to video k is described by $[i_k, L(i_k), s_{i_k}(u)]$, where $L(i_k)$ is the link set that is used by i_k and $s_{i_k}(u)$ is the group of users that access k from i. For multipath delivery scenarios, assuming client j accesses content via m interfaces and corresponding delivery rate of each interface f_k is $x_{i,j_{f_k}}$, applying **P2** only needs to rephrase the objective function of j to $J(\sum_{i=1}^{M} x_{i,j_{f_k}})$.

IV. Algorithm Design

In this section, we first decompose MMMP in terms of video clients and consider its dual problem. Then, we propose a distributed rate adaption algorithm DDA which supports individual clients to determine their optimal video bitrate.

A. Problem Decomposition

Considering **P2**, the objective function in eq. (5) is separable for- the video clients yet coupled by the constraints from eq. (6). In addition, as constraints in eq. (6) contain the maximum value function which is not differential, directly solving this problem is a nontrivial task. We introduce a new parameter x_i^l, and formulate the decomposed MMMP as follows:

U1: for each $x_{i,j}$

$$\max_{x_{i,j}\in[b_{\min},b_{\max}]} \quad J\left(x_{i,j}\right) \tag{7}$$

$$s.t \quad \sum_{k\in l(s)/i} x_k^l + x_{i,j} \le c_l, l\in p_j, i\in\mathcal{S}, j\in\mathcal{U} \tag{8}$$

$$x_{i,j} \le x_i^l, \quad l\in p_j, i\in\mathcal{S}, j\in s_i(u) \tag{9}$$

$\forall k\in\mathcal{S}, l\in\mathcal{L}$, x_k^l is defined as the video bitrate of i such that $x_k^l \ge x_{k,j}$ for all $j\in s(u)_l$. Constraint (8) indicates that for each link l used by j, the bitrate of j should not exceed the minimum residual link capacity, and eq. (9) says that the bitrate $x_{i,j}$ cannot exceed all x_i^l over its delivery path. We introduce following theorem.

Theorem 1: For each user $j\in\mathcal{U}$, the corresponding optimal value $x_{i,j}^*$ in **P2** can be derived equally by solving problem **U1**. Namely, $\forall i,j$, $x_{i,j}^*$ of **P2** and **U1** are equal.

Proof: See Appendix A. ∎

B. Distributed Optimal Rate Adaptation Algorithm

To derive the optimal $x_{i,j}^*$ of **U1** distributedly, we consider the dual problem of **U1**. Consider the Lagrangian of **U1**:

$$L_u\left(x_{i,j},\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j}\right) = J\left(x_{i,j}\right) - \sum_{l\in p_j}\upsilon_l\left(x_{i,j}-x_i^l\right) - \sum_{l\in p_j}\lambda_l\left(\sum_{j\in l(s)/i} x_j^l + x_{i,j} - c_l\right) \tag{10}$$

The Lagrangian dual function is thus:

$$D_u\left(\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j}\right) = \sup_{x_{i,j}\in[b_{min},b_{\max}]} L_u\left(x_{i,j},\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j}\right) \tag{11}$$

and the dual problem of **U1** can be formulated as follows:

$$\min_{\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j}\ge 0} \quad D_u\left(\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j}\right) \tag{12}$$

Since the optimal values of the primal and dual problems are equal due to the strong duality property of **U1**, the primal optimal solution $x_{i,j}^*$ can be recovered from a dual optimal point $(\boldsymbol{\lambda}_{\boldsymbol{p_j}}^*, \boldsymbol{\upsilon}_{\boldsymbol{p_j}}^*)$, namely:

$$x_{i,j}^* = \arg\max_{x_{i,j}\in[b_{\min},b_{\max}]} L_u\left(x_{i,j},\boldsymbol{\lambda}_{p_j}^*,\boldsymbol{\upsilon}_{p_j}^*\right)$$

Let $x_{i,j}(p_j)$ be the unique maximizer of $L_u(x_{i,j},\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j})$. If the inverse of $J(.)$ exists, according to the Karush-Kuhn-Tucker condition of **U1:A** [26], $x_{i,j}(p_j)$ can be derived by:

$$x_{i,j}\left(p_j\right) = J'^{-1}\left(\sum_{l\in p_j}(\lambda_l+\upsilon_l)\right) \tag{13}$$

As $D_u(\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j})$ is continuous and differential for $(\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j})$, the partial differentials of each λ_l, υ_l are:

$$\frac{\partial D_u}{\partial\lambda_l}\left(\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j}\right) = -\left(\sum_{k\in l(s)/i} x_k^l + x_{i,j} - c_l\right), l\in p_i \tag{14}$$

$$\frac{\partial D_u}{\partial\upsilon_l}\left(\boldsymbol{\lambda}_{p_j},\boldsymbol{\upsilon}_{p_j}\right) = -\left(x_{i,j}-x_i^l\right), \quad l\in p_i \tag{15}$$

Therefore, based on eq. (13), (14), (15), DDA solves the λ_l^* and υ_l^* of dual problem **U1:D** by gradient projection

method [27], and updates $x_{i,j}(t)$ iteratively, as follows:

$$x_{i,j}(t+1) \triangleq \begin{cases} b_{\min}, & \text{if } \lambda_l(t)+\gamma_l(t) > J'(b_{\min}) \\ b_{\max}, & \text{if } \lambda_l(t)+\gamma_l(t) < J'(b_{\max}) \\ J'^{-1}\Big(\sum_{l\in p_j}(\lambda_l(t)+\upsilon_l(t))\Big), & \text{otherwise} \end{cases} \tag{16}$$

$$\lambda_l(t+1) \triangleq \lambda_l(t) + \gamma\left(\sum_{k\in l(s)/i} x_k^l(t+1) + x_{i,j}(t+1) - c_l\right) \tag{17}$$

$$\upsilon_l(t+1) \triangleq \upsilon_l(t) + \gamma\Big(x_{i,j}(t+1) - x_i^l(t+1)\Big) \tag{18}$$

The above iterations suggest treating users, routers as processors in a distributed processing system, and the optimal rate of each client can be derived by only communicating with links over its delivery path, without coordination with other clients. Specifically, at each iteration t, client j solves $x_{i,j}(t)$ in eq. (16) by collecting $\lambda_l(t-1)$ and $\upsilon_l(t-1)$ from links over its delivery path p_j and communicates them the new derived $x_{i,j}(t)$. In parallel, client j requests video with bitrate $\arg\min_{b\in\mathbb{B}} \|x(t)-b\|_2$. Link l receives the $x_{i,j}(t)$ of all users that use l and select $\max_{j\in s_i(u)_l} x_{i,j}(t)$ as $x_i^l(t)$ for each source s in $l(s)$. Then, l uses all $x_k^l, k \in l(s)/i$ and $x_{i,j}$ to compute the $\lambda_l(t+1)$ and $\upsilon_l(t+1)$ by (17), (18). The derived $\lambda_l(t+1)$, $\upsilon_l(t+1)$ will be delivered to user j for computing the new $x_{i,j}(t+1)$ in the next iteration. The above process is repeated until the results reach the iteration criterion, $x_{i,j}(t+1) = x_{i,j}(t)$. $x_{i,j}(t)$, $\lambda_l(t)$, $\upsilon_l(t)$ are small enough and can be smuggled into *Interest* and *Data* packets, hence, do not require extra communication resources. The pseudocode of DDA is shown in **Algorithm 1**.

Convergence: assuming that initial $\boldsymbol{\lambda}(0)$ and $\boldsymbol{\upsilon}(0)$ are feasible, we have following convergence results.

Theorem 2: Given the utility function as (1), $-J''(x_{i,j}) \geq \frac{1}{\widetilde{\alpha}_j}$, where $\widetilde{a}_j > 0$, then, when step size γ satisfies $0 < \gamma < \frac{1}{\widetilde{A}\widetilde{L}\widetilde{S}}$, where $\widetilde{A} = \max_{j\in\mathcal{U}} \alpha_j$, from any initial point $\boldsymbol{x}(0)$, the $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\upsilon}^*)$ generated by **Algorithm 1** is dual optimal, namely, the $\boldsymbol{x}^*$ is the optimal adaptation rate for **P2**.

Proof: See Appendix B. ∎

Complexity: By observing **Algorithm 1**, the complexity of DDA at link side is mainly determined by the process. Let gradient projection iterates N times, and number of users and providers using link l are U_l and S_l, respectively. Thus, the complexity of algorithm at the link side is $O(N(U_l+S_l))$. At clients' side, the corresponding complexity is determined by the number of iteration of (16), which is N.

Time-varying adaptation: In order to extend DDA to the time-varying scenarios, the objective function **P2** can be reformulated as $f(\boldsymbol{x},t) = \sum_{i\in S(t)} \sum_{j\in s_i(u,t)} J(x_{i,j})$, where $s(t)$ and $s_i(u,t)$ are the set of providers and user set of provider i at time t, respectively. The $l(s)$ in constraint (6) is replaced by $l(s,t)$, which is the provider set that uses link l varying with t. Based on above changes, each end users still executes the same user algorithm as in **Algorithm 1**, except for computing the $p_j(t)$ in the place of p_j in (16). Each link executes the same link algorithm as in **Algorithm 1**, only with minor changes by replacing $l(s)$ in (17) with $l(s,t)$. Intuitively, if the change in link routings and providers is relative slower than the convergence rate, the algorithm still can converge to the optimal rates $\boldsymbol{x}^*$. We will further illustrate this feature by experimental tests in Section VI.

Algorithm 1: Distributed Delivery Algorithm (DDA) for Adaptive Streaming

Input: $\boldsymbol{x}(0), t=0$
Output: $\boldsymbol{x}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\upsilon}^*$
link l's algorithm:
while $\boldsymbol{\lambda}(t)! = \boldsymbol{\lambda}(t-1)$, $\boldsymbol{\upsilon}(t)! = \boldsymbol{\upsilon}(t-1)$ **do**
  receives the rate of $x_{i,j}(t)$ from all users that go through link l;
  foreach *provider i uses link l* **do**
    determines the $\max\{x_{i,j}(t) | j \in s(u)_l\}$;
  end
  foreach *user j goes through the link l* **do**
    computes the $\lambda_l(t)$, $\upsilon_l(t)$ according to (17), (18);
    communicates the $\lambda_l(t)$, $\upsilon_l(t)$ with user j;
  end
  t++;
end
$\boldsymbol{\lambda}^* = \boldsymbol{\lambda}(t)$, $\boldsymbol{\upsilon}^* = \boldsymbol{\upsilon}(t)$;
user j's algorithm:
while $x_{i,j}(t)! = x_{i,j}(t+1)$ **do**
  receives the sum of $\lambda_l(t)+\upsilon_l(t)$ from links in p_j;
  determines the next period delivery $x_{i,j}(t)$ by (16);
  requests video bitrate by $\arg\min_{b\in\mathbb{B}} \|x(t)-b\|_2$;
end
$x_{i,j}^* = x_{i,j}(t)$;
return $x_{i,j}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\upsilon}^*$;
final;

V. Heuristic Distributed Rate Adaptation Algorithm

The proposed DDA converges to the optimal value under any initial condition, as proved. However, the computation complexity of DDA at link grows with the number of passing users, which may trigger a scalability problem at the bottleneck links. In this context, in this section, we propose a lightweight heuristic distributed delivery algorithm (H-DDA).

Observe the following special case of **P2**:

$$\max_{\boldsymbol{x}\in[b_{\min},b_{\max}]^U} \quad \sum_{s_i\in\mathcal{S}} \log x_i \tag{19}$$

$$s.t \quad \sum_{i\in l(s)} x_i \leq c_l \quad l\in\mathcal{L} \tag{20}$$

The above problem describes a unicast scenario where each source serves one user only. The corresponding Lagrangian is:

$$L(\boldsymbol{x},\boldsymbol{\lambda}) = \sum_{s_i\in\mathcal{S}} \log x_i - \sum_{l\in\mathcal{L}} \lambda_l \left(\sum_{i\in l(s)} x_i - c_l\right)$$

This problem can be easily solved by considering its dual as in [21], similar to eq. (13),

$$x(p_i) = \left(\sum_{l \in p_i} \lambda_l \right)^{-1} \tag{21}$$

where p_i denotes the path used by provider i. Intuitively, the inverse of bitrate is equal to waiting delay of sending unit number of data according to the little's law.

For instance, when the data over link is 10Mbps, the delay of sending 1Mb data is 0.1s. Accordingly, $\sum_{l \in p_i} \lambda_l$ indicates the total delay of sending unit number of data using path i. Thus, the physical meaning of λ_l is the waiting delay over $\sum_{l \in p_i} \lambda_l$. Obviously, λ_l can be literately derived by following gradient projection method:

$$\lambda_l(t+1) = \left[\lambda_l(t) - \gamma \left(c_l - \sum_{i \in l(s)} x_i(t) \right) \right]^+ \tag{22}$$

Consider following problem with multicast feature:

P3:

$$\max_{\boldsymbol{x} \in [b_{\min}, b_{\max}]^U} \quad \sum_{s_i \in \mathcal{S}} \log x_i \tag{23}$$

$$s.t \quad \sum_{i \in l(s)} \max_{j \in s(u)_l} x_{i,j} \le c_l \quad l \in \mathcal{L} \tag{24}$$

Similar to eq. (22), given by the sending delay of l is only related to the load of link, we also have:

$$\lambda_l(t+1) = \left[\lambda_l(t) - \gamma c_l - \sum_{i \in l(s)} \max_{j \in s(u)_l} x_{i,j}(t) \right]^+ \tag{25}$$

for **P3**. Therefore, according to the physical meaning of λ_l,

$$x_{i,j}(t) = \left(\sum_{l \in p_j} \lambda_l(t) \right)^{-1} \tag{26}$$

extending eq. (26) to the generalized $J(.)$ which is strictly concave and continuous:

$$x_{i,j}(t) = J^{-1} \left(\sum_{l \in p_j} \lambda_l(t) \right) \tag{27}$$

Hence, at each iteration T of H-DDA, each link l in $\mathcal{G}$ collects the bitrate $x_{i,j}$ of clients over l, and determines the $\lambda_l(t)$ by (25). Link l communicates the $\lambda_l(t)$ to all users that use l. User j receives the $\lambda_l(t)$ of all l in p_j and calculates the $x_{i,j}(t)$. Links and users repeat this process until satisfy the stopping criterion of gradient method: for each l, at iteration T, $\lambda_l(T) - \lambda(T-1)$. The pesudocode of H-DDA is given in **Algorithm 2**.

According to the pseudocode of **Algorithm 2**, the computation complexity at link side is bounded by $O(N.S_l)$. Because $S_l \ll U_l + S_l$ in multicast scenario, **Algorithm 2** can significantly reduce the computation load at links.

Unlike the optimal convergence of DDA which was proved in Section IV, it is difficult to theoretically analyse the optimality of H-DDA. Instead, we use the physical meaning of $\boldsymbol{\lambda}$ of **P3** to explain how H-DDA approximates the optimal value. For each end user j, let $x^*_{i,j}$ be the corresponding optimal value derived by DDA, the inverse of $x^*_{i,j}$ is equal to the current waiting delay of path, say $\boldsymbol{\lambda}^*_p$. And because $\boldsymbol{\lambda}^*_p$ is the optimal delay of path which is equal to total sum of λ^*_l whose corresponding link l is in p_j. Therefore, because $\lambda_l(t)$ converges to the λ^*_l in H-DDA, hence $x_{i,j}(t)$ in H-DDA converges to the $x^*_{i,j}$ in DDA. We thereby prove the optimal approximation of H-DDA.

Algorithm 2: Heuristic Distributed Delivery Algorithm

Input: $\boldsymbol{x}(0), t = 0$
Output: $\boldsymbol{x}^*$, $\boldsymbol{\lambda}^*$
link l's algorithm:
while $\boldsymbol{\lambda}(t)! = \boldsymbol{\lambda}(t-1)$ **do**
  receives the rate of $x_{i,j}(t)$ from all users that go through link l;
  foreach *provider i* **do**
    determines the $\max_{j \in s(u)_l} x(i,j)$ for provider i
  end
  compute the $\lambda_l(t)$ according to (25);
  communicate the $\lambda_l(t)$ to all users over l;
  t++;
end
$\boldsymbol{\lambda}^* = \boldsymbol{\lambda}(t)$, $\boldsymbol{v}^* = \boldsymbol{v}(t)$;
user j's algorithm:
while $x_{i,j}(t)! = x_{i,j}(t+1)$ **do**
  receives the sum of $\lambda_l(t)$ from the links over its path;
  determines the next period delivery rate $x(t)$ according to (27);
  communicates the $x_{i,j}(t+1)$ to links $l \in p_j$;
  request video bitrate by $\arg\min_{b \in \mathbb{B}} \|x(t) - b\|_2$;
end
$x^*_j = x_{i,j}(t)$;
return $\boldsymbol{x}^*$, $\boldsymbol{\lambda}^*$;
final;

Furthermore, we also test the optimal approximation of H-DDA through numerical evaluation in MATLAB. We consider a tree-based network whose topology and link bandwidth are shown in Fig. 2. In this tree topology, four leaf nodes act as video clients continuously sending out DAS requests during the simulation.

Fig. 3 illustrates convergence of user rate of U1-U4 derived by H-DDA to the solutions of DDA. Observing that, both H-DDA and DDA converge to the same results, only different in convergence rate, hence permits using H-DDA to achieve the optimal rate adaptation.

Adoption to VR Applications: Our algorithm provides an adaptive streaming scheme and can be easily applied to the omnidirectional video such as VR. In VR, each image is splitted into multiple tiles and each tile is coded independently. When delivering adaptive VR streaming, viewport predictions are required to forecast the location of user's interested area. However, viewport prediction problem is beyond the scope of this manuscript. For more details of these methods, please refer to our previous work [23]. In our previous work [23],

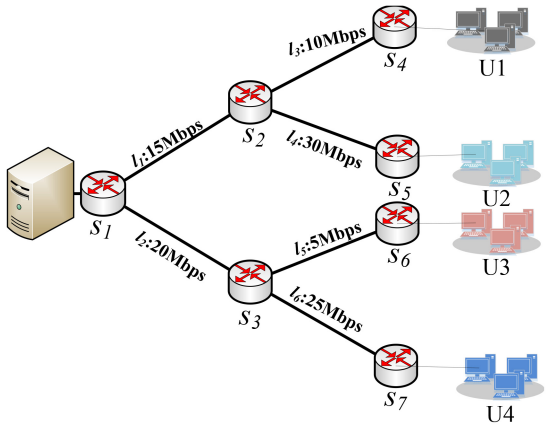


Fig. 2. Tree-based topology.

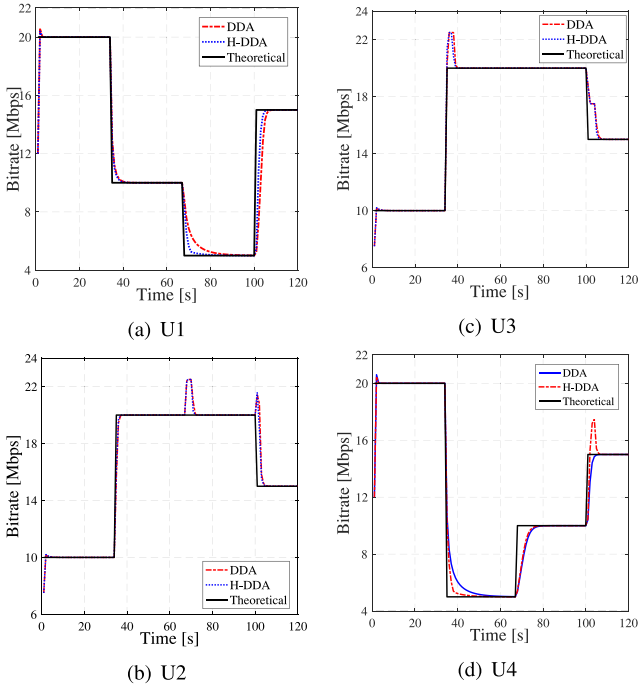


Fig. 3. User rate convergence comparison.

we propose a viewport prediction method which can derive the probability of a tile watched by a user. Let the probability of a tile v_i watched by a viewer be p_i and the optimal transmission rate is $x*$, for each tile, the allocated bandwidth can be given by:

$$x(v_i) = \frac{p_i}{\sum_{v_\eta}^{V}} x* \tag{28}$$

Then, v_i's bitrate b_i is:

$$b_i = \min_{b<x(v_i)} b - x(v_i)$$

The pseudocode is shown in **Algorithm 3**.

VI. Performance Evaluation

To evaluate the performance of the proposed algorithms, we implemented DDA and H-DDA in MATLAB and *NS*-3,

Algorithm 3: Rate Control for Omnidirectional Video

Input: x^*
Output: x^*, λ^*
foreach *video chunk* **do**
  invoke the viewport prediction algorithm to derive the viewing probability;
  foreach *tile* v_i **do**
    calculate the allocated bandwidth: $x(v_i) = p_i x* / \sum_{v_\eta}^{V}$;
    determine the bitrate for v_i: $b_i = \min_{b<x(v_i)} b - x(v_i)$;
  end
  deliver the tiles with bitrate $(b_1, b_2, ..., b_V)$;
end
final;

respectively. First, we present the simulation setup. Then, we analyse the convergence of DDA and H-DDA in time-varying condition and compare our algorithm against two state-of-art solutions HAVS [20] and DASH-BOLA [24].

A. Simulation Setup

We select BestRoute [8] as the request routing strategy, where routers maintain a routing table in order to discover replicas with minimum hop counts. For caching strategy, we employ the Leave Copy Everywhere (LCE) [8], which enables edge servers copying all passing content to their storage The size of the cache is randomly set to 25MB, 50MB and 100MB per router. For test videos, we use MPEG-DASH multimedia streaming with SVC-encoded format. The DASH video set is from [22], each segment is two seconds long and video set contains 8 movies with 120s of each. Each video is encoded into one base layer and four enhancement layers. The base layer b_1 has an average bitrate of 600kbps, and enhancement layers 1, 2, 3, 4 have 1600kps, 2600kps, 1940kps and 4440kps, respectively. To simulate the multicast scenario, a random number of users (from 1 to 5) will be selected to request the same video within a very same time window. The arrival rate of requests group follows the Poisson distribution with $\lambda = 0.05$. Each requests group randomly select a video to request by a Zipf distribution whose parameter is 0.8. After determining the video to ask, end users will request chunks of video in sequence and re-select a new video to request after requesting all chunks of current video.

B. Experimental Results

To simulate a realistic environment, we build a forest-based topology as in Fig. 4 which is widely used for Content Delivery Networks (CDN). The forest-based topology consists of 14 nodes and 13 nodes acting as video clients. To simulate the heterogeneous characteristics of an access network, the leaf routers act as access points (APs) with different communication technologies. For instance, AP1 and AP5 act as edge routers in wired networks, AP2 and AP4 are wireless access points which use the 802.11a protocol with 10 and 5Mbps

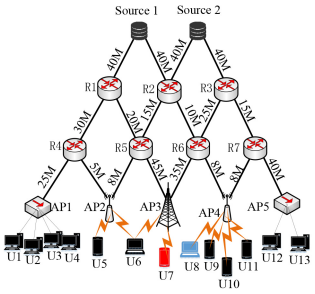


Fig. 4. Topology of forest-based network.

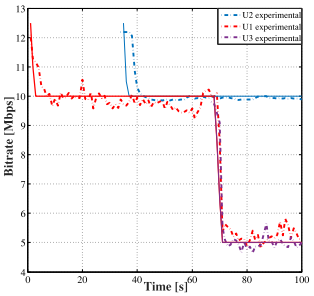


Fig. 5. Convergence analysis of U4-6 in Forest-based topology.

shared bandwidth, respectively. AP3 is a LTE network base station to simulate the cellular network environment which provides 4Mbps access bandwidth to each end user.

1) User Rate Convergence Analysis: Fig. 5 shows the rate convergence of $U4$, $U5$, $U6$ when accessing video from AP1 and AP2. The solid lines and dash lines correspond to the rate adaption provided by DDA and H-DDA, respectively. As the figure depicts, the simulation results of H-DDA converge well to the optimal value. Besides, the two algorithms also quickly adapt the network condition variation during the simulation. For example, when $U6$ begins to request video, the rate of $U6$ quickly decreases to the new optimal value 5Mbps, hence, showing the property of time-varying adaption of both DDA and H-DDA. Fig. 6 shows the convergence analysis of users accessing video from AP4, where all users share the access bandwidth with 5Mbps. As we expect, the rate of users at AP2 also converges well to the theoretical results. We also observe an interesting result in Fig. 6: both simulation and theoretical values show that when $U8$ $U9$ concurrently access DAS (at 22s during the simulation), the access bandwidth of WiFi is split into 2.5Mbps for each user, respectively. When more flows joining (At 35s and 75s), the bandwidth is further equally split into fours, which reveals the fairness of our algorithm. The above observation indicates that our proposed scheme can accommodate dynamic network variations. Additionally, note that a faster convergence can be achieved by relaxing the iteration criterion, but at the cost of larger upper bound of the convergence. This show that there is a tradeoff between the dynamic adaptation and better theoretical performance in DDA.

2) Average Bit Rate (ABR) Comparison: We define the ABR as the arithmetic mean of average bitrate of overall users. Fig. 7 show the ABR comparison of H-DDA, HAVS and

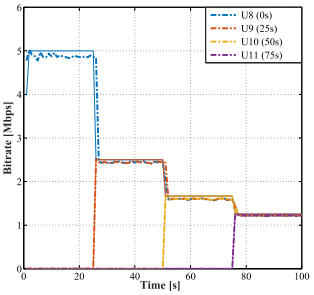


Fig. 6. Convergence analysis of U8-U11 in Tree-based topology.

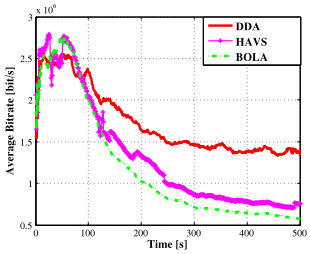


Fig. 7. ABR comparison of H-DDA, HAVS, BOLA.

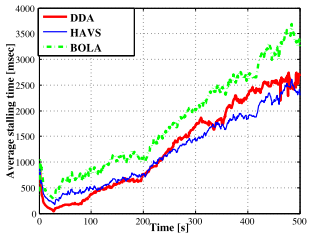


Fig. 8. AST comparison of H-DDA, HAVS, BOLA.

BOLA. As figure shows the ABR of three solutions experience a increasing trend at the beginning. After 50s, all solutions decrease and then enter periodical vibration phase. The red line corresponding to H-DDA achieves a 37% and 41% increment against the HAVS and BOLA. At the beginning, the network load is at low level and the links have enough bandwidth to support the requested high bitrate video. However, after the total bitrate reaches the link capacity limits, the continuous increase in video clients reduces the ABR. In H-DDA, the overall bitrate tracks to the theoretical optimal bound, hence, providing the best performance among three solutions. HAVS adjusts the data rate at each hop locally, which fails to optimize the user bitrate globally, and results in a relatively low bitrate against H-DDA. Regardless of the link capacity, each client in BOLA greedily requests higher bitrate video in order to maximize their own video quality, which may aggravate the network congestion when the network is already in a high load condition. Therefore, BOLA performs the worst.

3) Average Stalling Time (AST) Comparison: We define the time interval between playback freeze and restart as the stalling time. The shorter stalling time is, the smoother the playback experienced by the client is. We measure the average value of stalling time of using H-DDA, HAVS and BOLA and show the results in Fig. 8. We observe that the red curve corresponding to the H-DDA reduces the AST by 10% and

30% after 300s when comparing with HAVS and BOLA. As mentioned, H-DDA uses a distributed rate adaptive method to take full use of link bandwidth while avoiding the network congestion by limiting the total delivery rate to the link capacity, achieving a smoother playback. HAVS also limits the data rate to the link capacity at each hop, hence avoiding the network congestion and smooth playback at some level. However, the hop-level transmission control results in a sub-optimal rate control. BOLA uses a greedy method to request video content, which leads to higher risk of playback freeze and hence, it performs the worst among the three solutions.

VII. Conclusion and Future Works

This paper proposes a distributed optimal rate configuration algorithm for dynamic adaptive streaming. First the rate adaptation problem is formulated as MMMP, whose linear relaxation is concave. Then MMMP is decomposed in terms of video clients and DDA is proposed to enable users communicate optimally. Furthermore, a heuristic method named H-DDA which reduces the computation complexity in comparison with DDA, while maintaining the optimal approximation is introduced. Simulation results show algorithm convergence and illustrate how H-DDA outperforms other state-of-art solutions.

Although the theoretical proofs and simulations test validate the performance of our proposed algorithms, several open issues remain. First, our work focuses on wireless communications and it is necessary to consider the mobility of the nodes. Future work will study how to model the user mobility behavior and embed this into the design of our algorithm. Secondly, for live streaming services, transcoding the video content into multiple representations consumes large computation resources. Future research will jointly optimize the transmission and transcoding which are both critical to the high performance of 360 degree live streaming. Thirdly, future work will consider deploying the proposed algorithm in a real life environment and testing it.

Appendix A
Proof of the Theorem 1

Proof: Given by the definition of x_i^l, the problem **P2** can be rephrased as follows:

P2:A

$$\max_{\boldsymbol{x}\in[b_{\min},b_{\max}]^U} \quad \sum_{i\in\mathcal{S}}\sum_{j\in s(u)} J\left(x_{i,j}\right) \tag{29}$$

$$s.t \quad \sum_{i\in l(s)} x_i^l \le c_l \quad l\in\mathcal{L} \tag{30}$$

$$x_{i,j}\le x_i^l, \quad i\in\mathcal{S}, j\in s_i(u), l\in p_j, \tag{31}$$

We aggregate **U1** across all users and obtain:

U1:A

$$\max_{x_{i,j}\in[b_{\min},b_{\max}]} \quad \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)} J\left(x_{i,j}\right) \tag{32}$$

$$s.t \quad \sum_{k\in l(s)/i} x_k^l + x_{i,j} \le c_l, l\in p_j, i\in\mathcal{S}, j\in s(u) \tag{33}$$

$$x_{i,j}\le x_i^l, \quad l\in p_j, i\in\mathcal{S}, j\in s_i(u) \tag{34}$$

Assuming that $\boldsymbol{x}^*$ and $\boldsymbol{x}'^*$ are the optimal solutions of **P2:A** and **U1:A**, respectively, **Theorem 1** holds only when $\boldsymbol{x}^*=\boldsymbol{x}'^*$. Next, we show how to prove that $\boldsymbol{x}^*=\boldsymbol{x}'^*$.

Let the Lagrangian of **P2:A** and **U1:A** be as in eq. (35) and (36) shown at the top of the next page. The dual optimal values of eq. (35) and (36) are defined as $(\boldsymbol{x}^*, \boldsymbol{\lambda_p}^*, \boldsymbol{\upsilon}^*)$ and $(\boldsymbol{x}'^*, \boldsymbol{\lambda_p}'^*, \boldsymbol{\upsilon}'^*)$, respectively. According to the slackness complementarity condition [26], we have:

$$\begin{cases} \upsilon_{ijl}^* > 0, x_{i,j}^* = x_i^{l*} \\ \upsilon_{ijl}^* = 0, x_{i,j}^* < x_i^{l*}, \end{cases} \quad \begin{cases} \upsilon_{ijl}'^* > 0, x_{i,j}'^* = x_i^{l'*} \\ \upsilon_{ijl}'^* = 0, x_{i,j}'^* < x_i^{l'*}, \end{cases} \tag{37}$$

Using x^* to replace the x'^*, we have:

$$\sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j} \upsilon_{ijl}'^*\left(x_{i,j}^* - x_i^{l*}\right) = 0$$

For the case of $\sum_{k\in l(s)/i} x_k^{l*} + x_{i,j}^* < c_l$, the corresponding $\lambda_{ijl}'^* = 0$. This can be proved by contradiction. If there exists a $\lambda_{ijl}'^* > 0$, $x_k^{l*} + x_{i,j}^* < c_l$, $\lambda_{ijl}'^*(\sum_{k\in l(s)/j} x_k^{l*} + x_{i,j}^* - c_l) > 0$, this means it exists a $\hat{\boldsymbol{x}}^*$ such that

$$\sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)} J\left(\hat{x}_{i,j}^*\right) > \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)} J\left(x_{i,j}'^*\right)$$

which contradicts the assumption that $x_{i,j}'^*$ is the maximum value.

For case of $\sum_{k\in l(s)/i} x_k^{l*} + x_{i,j}^* = c_l$, we have $\sum_{l(s)/i} x_k^{l*} = \sum_{k\in l(s)/i} x_k^{l'*}$ and $x_{i,j}^* = x_{i,j}'^*$. Combining the above two cases, we have

$$\sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\lambda_{ijl}'^*\left(\sum_{k\in l(s)/i} x_k^{l*} + x_{i,j}^* - c_l\right) = \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\lambda_{ijl}'^*\left(\sum_{k\in l(s)/i} x_k^{l'*} + x_{i,j}'^* - c_l\right) \tag{38}$$

Let $g_{ijl}(\boldsymbol{x}'^*) = \sum_{k\in l(s)/i} x_k^{l'*} + x_{i,j}'^* - c_l$, given by eq. (38) and strong duality property of **U1:A** and **P2:A**, we have:

$$\begin{aligned} \sum_{i\in\mathcal{S}}\sum_{i\in s(u)} J\left(x_{i,j}^*\right) &= \sum_{i\in\mathcal{S}}\sum_{i\in s(u)} J\left(x_{i,j}^*\right) - \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\lambda_{ijl}'^* g_{ijl}(\boldsymbol{x}^*) \\ &\ge \sum_{i\in\mathcal{S}}\sum_{i\in s(u)} J\left(x_{i,j}'^*\right) - \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\lambda_{ijl}'^* g_{ijl}(\boldsymbol{x}'^*) \\ &= \sum_{i\in\mathcal{S}}\sum_{i\in s(u)} J\left(x_{i,j}'^*\right) \end{aligned} \tag{39}$$

$$L_p(\boldsymbol{x}, \boldsymbol{\lambda}_p, \boldsymbol{\upsilon}) = \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)} J(x_{i,j}) - \sum_{l\in\mathcal{L}}\lambda_l\left(\sum_{x\in l(s)} x_i^l - c_l\right) - \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\upsilon_{ijl}\left(x_{i,j} - x_i^l\right) \tag{35}$$

$$L_u(\boldsymbol{x}, \boldsymbol{\lambda}_u, \boldsymbol{\upsilon}) = \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)} J(x_{i,j}) - \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\lambda_{ijl}\left(\sum_{k\in l(s)/i} x_k^l + x_{i,j} - c_l\right) - \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\upsilon_{ijl}\left(x_{i,j} - x_i^l\right) \tag{36}$$

Similarly, we also have

$$\sum_{l\in\mathcal{L}}\lambda_l^*\left(\sum_{x\in l(s)} x_i^{l*} - c_l\right) = \sum_{l\in\mathcal{L}}\lambda_l^*\left(\sum_{x\in l(s)} x_i^{l'*} - c_l\right)$$

and $\sum_{i\in\mathcal{S}}\sum_{i\in s(u)} J(x_{i,j}^*) \le \sum_{i\in\mathcal{S}}\sum_{i\in s(u)} J(x_{i,j}'^*)$. Hence, $\sum_{i\in\mathcal{S}}\sum_{i\in s(u)} J(x_{i,j}^*) = \sum_{i\in\mathcal{S}}\sum_{i\in s(u)} J(x_{i,j}'^*)$. Since the optimal solution is unique given by the concave propriety of **P2:A**, $\boldsymbol{x}^* = \boldsymbol{x}'^*$, therefore **Theorem 1** holds. ■

APPENDIX B
PROOF OF THE THEOREM 2

Proof: Because **Algorithm 1** generates the $\boldsymbol{\lambda}(t)$,$\boldsymbol{\upsilon}(t)$ by the gradient projection method, hence, according to [27], $\boldsymbol{\lambda}(t)$,$\boldsymbol{\upsilon}(t)$ converges to $\boldsymbol{\lambda}^*$ and $\boldsymbol{\upsilon}^*$ only when ∇D_u is Lipschitz. Let $\beta(j) = \frac{1}{-J''(x_{i,j}(p_j))}$ and

$$A_j = \begin{bmatrix} B(j) & 0 \\ 0 & B(j) \end{bmatrix} = diag(\beta(j))_{2L\times 2L} \tag{40}$$

where each $B(j)$ is $L \times L$ matrix with diagonal elements $\beta(j)$. According to eq. (13), we have:

$$J''\left(x_{i,j}(p_l)\right)\frac{\partial x_{i,j}(p_{i,l})}{\partial p_{i,l}} = 1, p_{i,l} = \begin{cases} \lambda_l, & i = 1, l \in p_j \\ \upsilon_l, & i = 2, l \in p_j \end{cases} \tag{41}$$

Hence, $\frac{\partial x_{i,j}(p_{i,l})}{\partial p_{i,l}}$ can be represented as:

$$\frac{\partial x_{i,j}(p_{i,l})}{\partial p_{i,l}} = \frac{R_{lj}}{J''(x_i, j(p_l))}$$

where $R_{lj} \in \{0, 1\}$, $R_{lj} = 1$ indicates the user j go through link l and 0 otherwise. Using (41), we have vector

$$\left[\frac{\partial x_{i,j}(p_{i,l})}{\partial p_{i,l}}\right]_{2L} = -A_j C^T$$

where $C^T = [R, R]^T$, and $R = (R_{lj})$.

According to eq. (14) and eq. (15), we have:

$$\nabla^2 D_u(\boldsymbol{\lambda}, \boldsymbol{\upsilon}) = -C\left[\frac{\partial x_{i,j}(p_{i,l})}{\partial p_{i,l}}\right]_{2L}$$

and hence we have $\nabla^2 D_u(\boldsymbol{\lambda}, \boldsymbol{\upsilon}) = CA_jC^T$.

According to the mean value theorem, $\forall \boldsymbol{m}, \boldsymbol{n}$, we have

$$\begin{aligned}\nabla D_u(\boldsymbol{m}) - \nabla D_u(\boldsymbol{n}) &= \nabla^2 D_u(\xi)(\boldsymbol{m} - \boldsymbol{n}) \\ &= CA_j(\xi)C^T(\boldsymbol{m} - \boldsymbol{n})\end{aligned} \tag{42}$$

Given the Schwartz inequality property of 2-norm $\|.\|$,

$$\|\nabla D_u(\boldsymbol{m}) - \nabla D_u(\boldsymbol{n})\| \le \|CA_j(\xi)C^T\|.\|\boldsymbol{m} - \boldsymbol{n}\|$$

according to [27] (pp. 635).

$$\|CA_j(\xi)C^T\|^2 \le \|CA_j(\xi)C^T\|_\infty.\|CA_j(\xi)C^T\|_1$$

In particular, $\|(CA_j(\xi)C^T)'\|_\infty = \|CA_j(\xi)C^T\|_1$ and because $CA_j(\xi)C^T$ is symmetric, we further have $\|CA_j(\xi)C^T\|_\infty = \|CA_j(\xi)C^T\|_1$.

Therefore,

$$\begin{aligned}\|CA_j(\xi)C^T\|_2 &\le \|CA_j(\xi)C^T\|_\infty \\ &= \max_i \sum_j \sum_k \beta_k(w)R_{ik}R_{kj} \\ &= 2|p_j| \max_i \sum_k \beta_k(w)R_{ik} \\ &\le 2\widetilde{A}\widetilde{L}\widetilde{S}\end{aligned} \tag{43}$$

Therefore, ∇D_u is Lipschitz with

$$\|\nabla D_u(\boldsymbol{m}) - \nabla D_u(\boldsymbol{n})\| \le 2\widetilde{A}\widetilde{L}\widetilde{S}.\|\boldsymbol{m} - \boldsymbol{n}\|$$

Because the $J(.)$ is continuous and one-to-one mapping, $x_{i,j}(p_j)$ is continuous and therefore, $\lim_{t\to\infty} x_{i,j}(t) = x_{ij}^*$, hence, the theorem is proved. ■

REFERENCES

[1] G. Zhang and J. Y. B. Lee, "Ensemble adaptive streaming—A new paradigm to generate streaming algorithms via specializations," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1346–1358, Jun. 2020.

[2] A. Yaqoob and G.-M. Muntean, "A combined field-of-view prediction-assisted viewport adaptive delivery scheme for 360° videos," *IEEE Trans. Broadcast.*, vol. 67, no. 3, pp. 746–760, Sep. 2021.

[3] B. Wei, H. Song, S. Wang, and J. Katto, "Performance analysis of adaptive bitrate algorithms for multi-user DASH video streaming," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2021, pp. 1–6.

[4] C. Zhan, H. Hu, Z. Wang, R. Fan, and D. Niyato, "Unmanned aircraft system aided adaptive video streaming: A joint optimization approach," *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 795–807, Mar. 2020.

[5] X. Ma et al., "QAVA: QoE-aware adaptive video bitrate aggregation for HTTP live streaming based on smart edge computing," *IEEE Trans. Broadcast.*, vol. 68, no. 3, pp. 661–676, Sep. 2022.

[6] M. Yang, H. Liang, and F. Yang, "Real-time adaptive switching mechanism towards viewport-adaptive omnidirectional video streaming," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, 2021, pp. 1–6.

[7] M. Kim and K. Chung, "Edge computing assisted adaptive streaming scheme for mobile networks," *IEEE Access*, vol. 9, pp. 2142–2152, 2021.

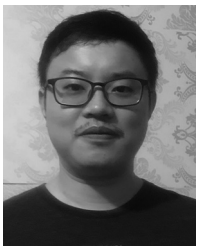
[8] P. Dai, F. Song, K. Liu, Y. Dai, P. Zhou, and S. Guo, "Edge intelligence for adaptive multimedia streaming in heterogeneous Internet of Vehicles," *IEEE Trans. Mobile Comput.*, vol. 22, no. 3, pp. 1464–1478, Mar. 2023.

[9] T. Feng, Q. Qi, J. Wang, J. Liao, and J. Liu, "Timely and accurate bitrate switching in HTTP adaptive streaming with date-driven I-frame prediction," *IEEE Trans. Multimedia*, early access, Apr. 6, 2022, doi: 10.1109/TMM.2022.3165381.

[10] A. Paul and S. Mitra, "Deep reinforcement learning based cooperative control of traffic signal for multi-intersection network in intelligent transportation system using edge computing," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 11, pp. 1–8, 2022.

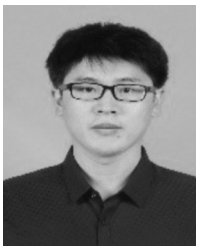
[11] N. Kan, J. Zou, C. Li, W. Dai, and H. Xiong, "RAPT360: Reinforcement learning-based rate adaptation for 360-degree video streaming with adaptive prediction and tiling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1607–1623, Mar. 2022.
[12] H. B. Salameh, H. Al-Obiedollah, T. Arabiat, A. Al-Ajlouni, and Y. Jararweh, "Joint bandwidth and power resource allocation technique in multi-carrier non-orthogonal multiple access-based cognitive Internet of Things networks," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 11, p. e4604, 2022.
[13] M. Shirmohamadi, H. Bakhshi, and M. Dosaranian-Moghadam, "Optimizing resources allocation in a heterogeneous cloud radio access network using machine learning," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 9, pp. 1–10, 2022.
[14] H. T. T. Tran, D. V. Nguyen, N. P. Ngoc, and T. C. Thang, "Overall quality prediction for HTTP adaptive streaming using LSTM network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3212–3226, Aug. 2021.
[15] H. Yuan, X. Hu, J. Hou, X. Wei, and S. Kwong, "An ensemble rate adaptation framework for dynamic adaptive streaming over HTTP," *IEEE Trans. Broadcast.*, vol. 66, no. 2, pp. 251–263, Jun. 2020.
[16] P. Lebreton and K. Yamagishi, "Predicting user quitting ratio in adaptive bitrate video streaming," *IEEE Trans. Multimedia*, vol. 23, pp. 4526–4540, 2021, doi: 10.1109/TMM.2020.3044452.
[17] A. Zhang et al., "Video super-resolution and caching—An edge-assisted adaptive video streaming solution," *IEEE Trans. Broadcast.*, vol. 67, no. 4, pp. 799–812, Dec. 2021.
[18] L. Zhong, X. Ji, Z. Wang, J. Qin, and G.-M. Muntean, "A Q-learning driven energy-aware multipath transmission solution for 5G media services," *IEEE Trans. Broadcast.*, vol. 68, no. 2, pp. 559–571, Jun. 2022.
[19] L. Cui, D. Su, S. Yang, Z. Wang, and Z. Ming, "TCLiVi: Transmission control in live video streaming based on deep reinforcement learning," *IEEE Trans. Multimedia*, vol. 23, pp. 651–663, 2021, doi: 10.1109/TMM.2020.2985631.
[20] Z. Liu and Y. Wei, "Hop-by-hop adaptive video streaming in content centric network," in *Proc. IEEE Conf. Commun (ICC)*, 2018, pp. 1–7.
[21] N. Eswara, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya, "Perceptual QoE-optimal resource allocation for adaptive video streaming," *IEEE Trans. Broadcast.*, vol. 66, no. 2, pp. 346–358, Jun. 2020.
[22] C. Kreuzberger, D. Posch, and H. Hellwagner, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP," in *Proc. ACM MMSys*, Portland, OR, USA, 2015, pp. 213–218.
[23] M. Wang, S. Peng, X. Chen, Y. Zhao, M. Xu, and C. Xu, "CoLive: An edge-assisted online learning framework for viewport prediction in 360° live streaming," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2022, pp. 1–6.
[24] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1698–1711, Aug. 2020.
[25] Y. Liu and J. Y. B. Lee, "A unified framework for automatic quality-of-experience optimization in mobile video streaming," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–7.
[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[27] D. P. Betsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

Lujie Zhong received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2013. She is currently an Associate Professor with the Information Engineering College, Capital Normal University, Beijing. She has published papers in prestigious international journals and conferences in the related area, including *IEEE Communications Magazine*, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE INTERNET OF THINGS JOURNAL, IEEE INFOCOM, and ACM MM. Her research interests include communication networks, computer system and architecture, and mobile networks.



Mu Wang (Member, IEEE) received the Ph.D. degree in computer technology from the Beijing University of Posts and Telecommunications, China, in 2020. He currently serves as an Associate Researcher with the State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include information centric networking, wireless communications, and multimedia sharing over wireless networks.

Changqiao Xu (Senior Member, IEEE) received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences in January 2009. He was a Researcher with the Athlone Institute of Technology and a Joint Training Ph.D. with Dublin City University, Ireland, from 2007 to 2009. He joined Beijing University of Posts and Telecommunications, China, in December 2009, where he is currently a Professor with the State Key Laboratory of Networking and Switching Technology and Director of the Network Architecture Research Center. He has edited two books and published over 200 technical papers in prestigious international journals and conferences, including *IEEE Communications Magazine*, IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON MOBILE COMPUTING, INFOCOM, and ACM Multimedia. His research interests include network security, mobile networking, multimedia communications, and future Internet technology. He has served a number of international conferences and workshops as the co-chair and a TPC member. He is currently serving as the Editor-in-Chief of *Transactions on Emerging Telecommunications Technologies* (Wiley). He has received the National Natural Science Funds for Distinguished Young Scholar.



Shujie Yang received the Ph.D. degree in computer technology from the Beijing University of Posts and Telecommunications, China, in 2020. He currently serves as an Associate Professor with the State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include information centric networking, wireless communications, and multimedia sharing over wireless networks.

Gabriel-Miro Muntean (Fellow, IEEE) is a Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and the Co-Director of DCU Performance Engineering Laboratory. He has published four books and over 450 papers in top international journals and conferences. His research interests include rich media delivery quality, performance, and energy-related issues, technology enhanced learning, and other data communications in heterogeneous networks. He is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, the Multimedia Communications Area Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and reviewer for important international journals, conferences, and funding agencies. He coordinated the EU project NEWTON and leads the DCU team in the EU project TRACTION.