A Novel Distributed Multi-Source Optimal Rate Control Solution for HTTP Live Video Streaming

Shujie Yang, Chuxing Fang, Lujie Zhong, Mu Wang, Zan Zhou, *Member, IEEE*, Han Xiao, *Member, IEEE*, Hao Hao, Changqiao Xu, *Senior Member, IEEE*, and Gabriel-Miro Muntean, *Fellow, IEEE*

***Abstract*—HTTP live streaming delivers dynamically video content with varying bitrates to accommodate the dynamic real-time bandwidth fluctuations while considering diverse user preferences and device capabilities. Existing flow control solutions do not provide support for new features such as multi-source content transmission. In this paper, we propose a distributed multi-source rate control optimization algorithm (DMRCA) that maximizes the overall network bandwidth utility and improves viewer Quality of Experience (QoE). First, we model the rate control problem as a dual-optimized multi-source and multi-rate problem. Then, we decompose the problem into sub-problems of source rate selection and user rate adaptation and we prove that solving the original problem is equivalent to solving these two sub-problems. Furthermore, we propose DMRCA as a fully distributed algorithm to solve these sub-problems and derive an optimal solution and we discuss DMRCA's complexity and convergence. Finally, through a series of simulation tests, we demonstrate the superiority of our proposed algorithm compared to alternative state-of-the-art solutions.**

***Index Terms*—Rate control, HTTP live streaming, dual optimization theory.**

Manuscript received 18 January 2024; revised 13 March 2024; accepted 19 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62225105, Grant 62394323, Grant 62301070, Grant 92167204, Grant 62072030, and Grant 62394322; in part by the Zhe Jiang Lab Open Research Project under Grant K2022QAOAB05; in part by the Postdoctoral Science Foundation of China under Grant 2022M720518; in part by the Beijing Natural Science Foundation under Grant 4244084, and in part by the Science Foundation Ireland under Grant 12/RC/2289_P2 (Insight) and Grant 21/FFP-P/10244 (FRADIS). *(Corresponding authors: Changqiao Xu; Mu Wang.)*

Shujie Yang, Chuxing Fang, Mu Wang, Zan Zhou, Han Xiao, and Changqiao Xu are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100000, China (e-mail: sjyang@bupt.edu.cn; fcxfcx@bupt.edu.cn; muwang@bupt.edu.cn; zanezanzhou@gmail.com; xiaohan@bupt.edu.cn; cqxu@bupt.edu.cn).

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

Hao Hao is with the Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250202, China, and also with the Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan 250014, China (e-mail: haoh@sdas.org).

Gabriel-Miro Muntean is with the Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Dublin 9, D09 V209, Ireland (e-mail: gabriel.muntean@dcu.ie).

Digital Object Identifier 10.1109/TBC.2024.3391051

I. Introduction

LATELY, there is a surge in the popularity of high quality multimedia streaming over the Internet [1], [2], [3], [4], enabled by its ability to provide timely, immersive, and personalized viewing experience. The video streaming market is forecasted to grow by $310.44 bn during 2022-2027, accelerating at a CAGR of 20.36% during the forecast period [5]. The widespread adoption of smart devices and the heterogeneity of communication technologies (e.g., 5G and beyond, WiFi, Fiber) have offered video users various resolution capabilities, screen sizes, processing power and diverse access network bandwidth and delay, among other aspects [6]. These support the provision of an unprecedented range of services to an increasing video viewing population. Diverse solutions were proposed which tailored video content to different presentations (i.e., bit-rate, resolution, etc.) and dynamically performed adaptive delivery to meet bandwidth fluctuations, diversity of device characteristics or other deployment-related requirements. For instance, the authors of [7] have proposed an user gaze-driven adaptive solution for omnidirectional video delivery, researchers of [8] have designed an energy-aware adaptive solution based on machine learning and QAVA [9] has introduced a quality-aware adaptive video bitrate solution based on smart edge computing. DQAMLearn [10] proposed a solution for educational video quality control on mobile devices with different features, authors of [11] discussed an innovative solution for virtual reality video streaming and researchers of [12] proposed a viewport reconstruction-based 360° video caching solution for Tile-adaptive streaming. Authors of [13] have introduced a decentralised multicast adaptive solution, those of [14] have proposed a fuzzy logic solution for adaptive video streaming and researchers of [15] have proposed a new neural enhanced adaptive streaming framework for variable bit rate encoded videos, which is based on selective prefetching of video blocks. Among adaptive video streaming solutions, some researcher focus on HTTP Live Streaming [16], [17], [18], which supports users to enjoy high quality video streaming through the HTTP protocol. These solutions provide lightweight but effective algorithms that can be easily deployed in Web video applications. HTTP Live Streaming has emerged as a promising paradigm for video streaming in the context of future networks and services.

However, most studies related to HTTP live video streaming concentrate on how to allocate network bandwidth given

a certain target bitrate suggested by an Adaptive Bit-rate (ABR) algorithm [19], [20], [21]. While most ABR algorithms address the bit-rate selection issue at the client side, little attention is given to the rate control problem at the video provider's end. Nevertheless, it should be noted that this rate control issue directly impacts bandwidth utilization of links and improving network bandwidth utilization enables nodes within the network to transmit data at higher rates. As a result, there is a direct improvement in video transmission quality, which is desired. Unfortunately, most rate control algorithms do not consider key aspects of video transmission scenarios such as multi-source topology and multi-rate transmission [23], [24], [25], [26].

The consideration of these new aspects increases the complexity of both the solutions and network topology. A higher number of video source providers allows for content delivery by multiple node servers within the network, resulting in the possibility of a serving node switch during transmission, and thereby altering the transmission path [22]. This makes it more challenging for servers to access comprehensive network information for performance improvement purposes. For instance, they may be able to utilize the information associated with their own node only for flow control. Therefore, there is a need to propose a distributed rate control strategy that specifically considers multi-source and multi-rate aspects during the video transmissions, in order to enhance network bandwidth utilization and optimize user experience.

In this context, this paper proposes an innovative distributed HTTP Live Streaming rate control mechanism, based on a mathematical investigation of the achievable global rate optimization in a multi-source multi-rate video delivery context. An important feature of the proposed **distributed multi-source rate control optimization algorithm (DMRCA)** is that it enables providers and video consumers to dynamically select optimal streaming rates individually, without relying on a centralized control. The effectiveness of the proposed algorithm is assessed via extensive simulation tests in comparison with alternative approaches.

In summary, this paper's main contributions are as follows:

1) *Multi-source rate-adaptive problem*: by theoretically modeling the HTTP Live Streaming, we formulate the optimal rate selection problem of HTTP Live Streaming as a multi-source rate-adaptive problem (MAP). We then introduce a linear relaxation of MAP and prove its concavity, which has unique optimal solutions.
2) *Problem decomposition*: MAP is decomposed into a rate selection sub-problem (SRSP) and an user rate adaptive sub-problem (URAP). We prove the equivalence between the original MAP and the two sub-problems. Furthermore, we discuss the duality of the two sub-problems in order to solve the rate control problem in a decentralized context.
3) *Distributed design*: we propose a distributed optimization for the DMRCA algorithm to derive the global optimal rate control which enables providers and users to solve the rate control problem without central coordination. Moreover, the complexity, convergence and time varying adaptation aspects of DMRCA are also discussed.
4) *Performance evaluation*: DMRCA was evaluated against other state-of-art solutions under different network topologies. The simulation tests show how the proposed algorithm outperforms existing solutions in terms of average bitrate and playback freeze frequency.

The rest of the paper is organized as follows: Section II surveys related works. Section IV describes the network and QoE models. Section V analytically formulates MAP and its linear relaxation, and the problem is decomposed into two sub-problems in Section VI. The DMRCA design is carried out in Section VII. Performance evaluation and conclusions are provided in Sections VIII and IX, respectively.

II. Related Works

The stringent demands for high throughput and low latency associated with the latest video services dictate significant requirements in relation to the transmission rates. Diverse flow control mechanisms are employed to determine the rate at which data packets should be transmitted. By optimizing the flow control process, video content can be transmitted at a higher rate and lower loss, thereby enhancing the quality of presentation. This is equally performed for pre-recorded and live video streaming and the goal is to improving the viewers' Quality of Experience (QoE). Consequently, numerous research efforts have been focused on this issue, aiming to design innovative flow control algorithms to improve the performance of content delivery, including by maximizing the overall network bandwidth utilization.

In [23], a throughput control method that leverages the HTTP2 Flow Control mechanism is proposed. The author designs a video streaming framework, in which a manager is situated at the client side. This manager continuously monitors the bandwidth of the bottleneck network and manages the throughput by adjusting the flow control window size. The authors note that while some solutions utilize the TCP flow control to manage server sending rates, this approach may be susceptible to security concerns. Given that the flow control method in [23] is deployed at the application layer, it is considered to be a relatively straightforward solution to deploy and upgrade.

In the context of recent protocols such as QUIC, due to its implementation based on UDP, which itself does not have flow control, it implements its own native flow control strategy. The flow control of QUIC itself is based on restrictions, which mainly include two parts. The first part is to limit the amount of data that can be sent on each flow to prevent a single flow from occupying the entire receiving buffer of the connection. The second part is to limit the total number of bytes of stream data sent in stream frames across all streams. An enhanced flow control algorithm has been proposed to improve the original credit-based algorithm [24]. This new algorithm modifies the threshold one packet ahead of the original flow control algorithm in QUIC. This change circumvent misjudgments of the flow control update signal's timing. Concurrently, the

method for updating the maximum receive offset has been altered, thereby avoiding the potential sub-optimal behavior of the original scheme. The enhanced algorithm has been validated through simulation experiments and has been demonstrated to achieve the optimum performance in FC-limited scenarios.

Recently, machine learning techniqueswere employed to solve increasingly complex problems, including to address rate control issues. Iris [25] introduced an end-to-end statistical learning-based congestion control method for real-time video transmission. Within Iris, all streams maintain a small, fixed number of packet queues to ensure minimal latency and equitable bandwidth allocation. For rate control, a statistical learning approach is employed to dynamically adjust the transmission rate via online linear regression learning. This eliminates the need for the fixed-step adjustment strategy employed in traditional methods, resulting in accelerated convergence. In [26], a reinforcement learning-aided in-network congestion control scheme was proposed to address network volatility on a time scale of 10 to 100 milliseconds. The algorithm is directly deployed at the switches. The scheme leverages a multi-agent deep deterministic policy gradient (MADDPG) algorithm to enable a distributed execution during centralized training. A component within the training center gathers information regarding the behavior of all switches in the system and provides feedback, enabling the switches to complete the training process in a distributed manner.

In the field of dynamic bit rate encoded videos, some flow control strategies limit the amount of data from the encoding level. In recent research of [27], a 360 degree video encoding rate control (RC) algorithm based on virtual competitors is proposed, which combines game theory to propose a frame level bit allocation model based on virtual competitors. The algorithm provides a GOP level bit allocation scheme and designs an overall bit rate allocation scheme based on this to reduce the bit rate fluctuation of GOP. The scheme has been proven to have the optimal Rate Control error and bit rate fluctuation. Authors in [28] have proposed a quality control algorithm to replace the rate control algorithm in video transmission. The algorithm models the optimization problem of rate distortion based on the rate distortion model, and solves this continuous convex problem through the Karush-Kuhn-Tucker equation. The author combines the rate quality (R-Q) model to implement the proposed algorithm in versatile video coding (VVC), and conducts verification experiments to prove that the algorithm can achieve stable coding quality while ensuring coding performance.

However, multi-source capabilities represent a novel feature that cannot be overlooked when addressing flow control challenges. Performance-aware multi-source delivery solutions would benefit from information about the network. However, there is a lack of access to comprehensive network information, particularly needed when confronted with the increased complexity brought about by multi-source attributes. Implementing flow control algorithms across all nodes to achieve optimal overall bandwidth utilization poses significant difficulties. Consequently, distributed flow control algorithms are required to enable nodes to make rate decisions for global optimization in the absence of information from other nodes. To date, to the best of the authors' knowledge, previous research on the flow control problem in video streaming [4], [23], [24], [25], [26], [27], [28] has yet to demonstrate in practice the benefit of distributed algorithms. Therefore, an urgent need exists for a distributed rate control method that not only supports the multi-source feature but also provides optimal rate configuration for HTTP live streaming.

III. Distributed Framework

In the context of HTTP live streaming, users can access network services using a variety of devices and from various locations, as illustrated in Figure 1. The figure shows how some users employhigh end computing devices connected to the network via wired connections, thereby enjoying higher bandwidth. Conversely, other users leverage mobile devices to access the network through WiFi access points (AP) or 5G base stations (BS), resulting in comparatively lower bandwidth at the client side.

Concurrently, within the network, several provider nodes cache video resources. On the cloud side, high-performance resource servers are interconnected with edge servers situated closer to users through the core network. These servers are equipped with the capability to cache video resources and provide video transmission services. Along the link between edge servers and users, there may be several router nodes involved. The provider nodes offer various video bit-rates to cater to diverse network conditions and user preferences. For instance, when users encounter poor client-side network conditions leading to video stuttering or delays, they may opt to reduce the bit-rates to ensure smooth playback.

However, user nodes are unable to obtain the status information of the entire network. Although clients can estimate network congestion by calculating in real-time the network bandwidth, their source of information is limited to the links connected to the client. Information regarding congestion at other network nodes, such as congestion at router nodes and server loads, is inaccessible to the client. This makes it difficult determining the appropriate video rate to be requested. A similar information gap exists at service nodes; service nodes can also gather network-related information from the multiple links connected to them. However, within a large network topology, a single node remains unable to ascertain the overall network congestion. As a result, the provider nodes also face the issue of what rate they need to transmit the video data. In the context of the illustration from Figure 1, we consider a distributed framework in which each node deploys a distributed flow control algorithm such as the proposed DMCRA that determines the optimal bit rate selection solely based on information received from the connected link.

IV. System Model

In this section, we present the network and QoE models to describe the HTTP live streaming system mathematically. Table I summarizes the notations used in the rest of the paper.

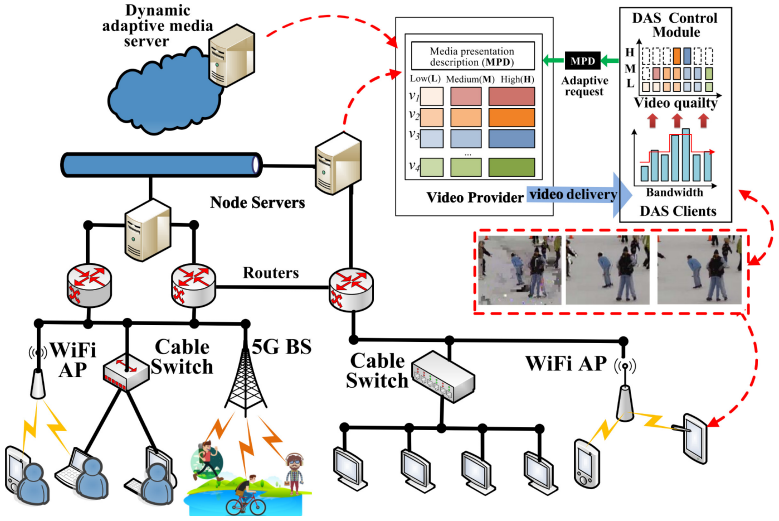


Fig. 1. Distributed HTTP Live Streaming Framework.

TABLE I
NOTATIONS

Symbol	Description
$\mathcal{G}$	The graphical topology of network
$p(i,j)$	The link set of path from j to i
$l_{x,y}$	One-hop link from node x to y
$\mathcal{S}$	The set of content providers in $\mathcal{G}$
$\mathcal{U}$	The set of video users in $\mathcal{G}$
$\mathcal{L}$	The set of links in $\mathcal{G}$
$\mathbb{B}$	The types of representations of HLS
$b_{\max}, b_{\min}$	The maximum and minimum bitrates of HLS
$L(i)$	The link set that used by provider i
$s_i(u)$	The user set of provider i
$s_j(u)_l$	The group of users of provider j that use link l
$D(b_m)$	The data size of video segment under quality b_m
T	The playback time of video segment

A. Network Model

We consider a network of N nodes including source servers, node servers and users that communicate with each other over a given connected, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ and $\mathcal{L} \subseteq \mathcal{V} \times \mathcal{V}$ denote the set of nodes and links between nodes, respectively. Let $\mathcal{S} = \{1, \ldots, S\} \in \mathcal{V}$ be set of video providers in the network and $\mathcal{U} = \{1, \ldots, U\} \in \mathcal{V}$ the set of end users. Due to the in-network caching, we assume that all the node servers are equipped with content repositories and thereby they can be also treated as video providers, namely, they belong to set $\mathcal{S}$. Let $p(i,j)$ denotes the order set of links of the path between the end user j and its provider i:

$$p(i,j) \triangleq \left\{ l_{i,x_1}, l_{x_1,x_2}, l_{x_2,x_3}, \ldots, l_{x_n,j} \right\}$$

where $i, x_k \in \mathcal{S}, k = 1, 2, 3, \ldots, n$ and $j \in \mathcal{U}$. $l_{x,y}$ indicates the link between x and y. Assume the network is connected (i.e., all users in $\mathcal{U}$ are able to access content from any provider in $\mathcal{S}$).

In our model, we consider the content is encoded by scaled video coding (SVC) [29] which encodes video into a base layer and several enhancement layers. Thus, video can either be decoded with only the base layer or with base and multiple enhancement layers; the more enhancement layers decoded, the better quality of video can be achieved. Let the video in $\mathcal{G}$ consist of one base layer and m enhancement layers, the bit-rates of the base layer and each enhancement layer k are b_1 and h_k, respectively. Accordingly, the types of representation of video v can be defined as $\mathbb{B} \triangleq (b_1, b_1 + h_1, b_1 + h_1 + h_2, \cdots, b_1 + h_1 + \cdots + h_m)$, and let $b_{\min} = b_1$ and $b_{\max} = b_1 + h_1 + \cdots + h_m$. To simplify the description, we assume that the required transmission rate of the video is equal to its playback bit-rates and all videos in $\mathcal{G}$ have equal $\mathbb{B}$.

B. QoE Model

Currently, most adaptive streaming services support the HTTP-based DASH protocol [30]. Along the traditional HTTP streaming, diverse video performance metrics are considered in DASH-based adaptation, including bitrate, stalling and startup delay. Consequently, diverse QoE models have been proposed for DASH with diverse explanations; their use may impact differently the final QoE estimation results. We extend the QoE model introduced in [31] according to our problem scenario:

$$QoE = \max\left(\frac{5.67x}{x_{\max}} + 0.17 - 4.95F, 0 \right) \quad (1)$$

where x is the chosen bitrate and $x_{\max}$ is the transmission rate of the best content representation $b_{\max}$. The factor F calculates the impact of stalling time and increases with the increase of the jamming frequency. The detailed calculation method is described in [31]. In a HTTP live video streaming scenario, the occurrence of stalling is related to the bitrate. With the increase of bitrate, the possibility of congestion in the network gradually increases. This leads to higher frequency of stalling and lower QoE. However, the sending bitrate of the provider is not able to directly determine the stalling time of the video at the consumer side. In order to minimize the stalling time, we can control the bitrate to meet the following condition:

$$\frac{D(b_m)}{x} < T$$

where for a certain video quality level b_m, the corresponding video segment data size is D_{b_m} and T denotes the time length of the video segment. This condition indicates that the sending time of the video segment should be less than its playback time. In this situation, the impact of stalling can be minimized and F can be regarded as a constant. Thus, we will focus on the rate control problem which mainly considers how to select the playback bitrate to optimize the network bandwidth utility and user QoE.

V. Problem Formulation

Let x_j be user j's delivery rate and $\forall j \in \mathcal{U}, x_j \in \mathbb{B}$. We refer to the vector $\boldsymbol{x} = \{x_1, x_2, x_3, \ldots, x_U\} \in \mathbb{B}^U$ as the *network rate configuration*. Generally, we assume that $\mathcal{S} = \bigcup_{i\in\mathcal{S}} s_i(u)$ and $s_i(u) \cap s_{i'}(u) = \phi, \forall i, i' \in \mathcal{S}$. $\boldsymbol{x}$ can be also rephrased as $\{x_{1,1}, \ldots, x_{i,j}, \ldots, x_{S,U}\}$, where $x_{i,j}$ implies the delivery rate of user j that receives the video from provider i. We represent the capacity of links in $\mathcal{L}$ in terms of a vector $\boldsymbol{c} = \{c_1, c_2, c_3, \ldots, c_L\}$. The rate configuration problem of HTTP live video streaming can be described as follows: given a utility function $f(\boldsymbol{x})$ varying with rate configuration $\boldsymbol{x}$, how to select a $\boldsymbol{x}^*$ so that $f(\boldsymbol{x})$ is maximized under the link capacity constraints $\boldsymbol{c}$.

In this paper, $f(\boldsymbol{x})$ is defined as the overall sum of user QoE values, as follows:

$$f(\boldsymbol{x}) = \sum_{i\in\mathcal{S}} \sum_{j\in s(u)} J\big(x_{i,j}\big)$$

where $J(\cdot)$ is the QoE estimation function using the model described in eq. (1). Int hsi context, we refer to the rate configuration of HTTP live video streaming as a multi-source adaptive rate problem (MAP), which is outlined below:

P1:

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

$$s.t. \sum_{i\in l(s)} x_{i,j} \le c_l - b_l \quad l \in \mathcal{L} \tag{3}$$

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

$$\frac{D\big(b_{i,j}\big)}{x_{i,j}} \le T \tag{5}$$

where $l(s)$ denotes the set of video providers that use link l. The inequalities in constraints (3) ensure that for any link l, the total rate of providers that use l cannot exceed the capacity c_l. Constraints (4) indicate the possible delivery rate of each user j is in $\mathbb{B}$ and constraints (5) control the bitrate to minimize the impact of stalling. Considering the form of (4), MAP is an integer programming problem which may be hard to solve. Instead, we consider a linear relaxation version of the MAP, which can be represented as follows:

P2:

$$\max_{\boldsymbol{x}\in[b_{\min},b_{\max}]^U} \sum_{s_i\in\mathcal{S}} \sum_{i\in s(u)} J\big(x_{s,j}\big) \tag{6}$$

$$s.t. \sum_{i\in l(s)} \max_{j\in s_i(u)_l} x_{i,j} \le c_l \quad l \in \mathcal{L} \tag{7}$$

$$\frac{Db_{i,j}}{x_{i,j}} \le T \tag{8}$$

where $\boldsymbol{x} \in [b_{\min}, b_{\max}]^U$ indicates the rate configuration which can be chosen from a continuous U-dimensional close space, which is considered as the relaxation of constraint (4) in **P1**. Particularly, the following theorem holds for **P2**:

Theorem 1: Given the $J(x)$ is concave and twice differential as eq. (1), the problem **P2** is a concave optimization problem, namely there exists a unique rate configuration $\boldsymbol{x}$ which maximizes the (6) under constraint (7).

Proof: Intuitively, $\sum_{s\in\mathcal{S}} \sum_{i\in s(u)} J(x_{s,j})$ is concave given that $J(.)$ is concave and concave propagation propriety of the summation [32]. For $\forall \boldsymbol{x}, \boldsymbol{y} \in [b_{\min}, b_{\max}]^U$ and $0 < \theta < 1$, we have:

$$\theta\boldsymbol{x} + (1-\theta)\boldsymbol{y} \in [b_{\min}, b_{\max}]^U$$

This is because for $\forall i$-th component of $\boldsymbol{x}$ and $\boldsymbol{y}$, x_i, y_i belong to the continuous interval $[b_{\min}, b_{\max}]$, we apparently have:

$$\theta x_i + (1-\theta) y_i \in [b_{\min}, b_{\max}]$$

Thus, the closure space $[b_{\min}, b_{\max}]^U$ is a convex set. $\max_{j\in s_i(u)_l} x_{i,j}$ is a maximum function which is convex [32]. Therefore, the relaxation version of MAP is a concave optimization problem [32] and has an unique $\boldsymbol{x}^*$ that globally optimizes **P2**. ■

Remark: In spite of focusing on a network with the multi-source feature, the relaxed MAP can be also easily generalized to other scenarios with minor modifications:

1) *Provider with Multiple Video Flows Scenario*: Instead of only delivering one video as we assumed in Section IV, providers in realistic environments may concurrently serve multiple videos. To be able to apply our proposed approach in this case, we split the provider i with n video flows into n virtual sources, each virtual source corresponding to a video flow and can be represented by following 4-tuple:

$$\big[i_k, L(i_k), s_{i_k}(u), b_{\min}, b_{\max}\big]$$

where i_k indicates the virtual source corresponding to flow k, $L(i_k)$ is the link set used by video flow k and $s_{i_k}(u)$ the group of users that access k from i.

2) *Multi-path Scenario*: A typical network in realistic environments also supports multi-path delivery. In this scenario, assuming end user j accesses content from $[f_1, f_2, \ldots, f_M]$ video sources, the corresponding delivery rate of each source f_k is $x_{i,j_{f_k}}$. Hence, the total delivery rate of user j:

$$x_u = \sum_{k=1}^{M} x_{i,j_{f_k}}$$

Then, the QoE function of u can be written as:

$$J\left(\sum_{i=1}^{M} x_{i,j_{f_k}}\right).$$

VI. Problem Decomposition

The distributed aspect of the network and the large numbers of end users make it difficult to maintain a central controller to configure the delivery rate for all end users. Consequently, a distributed method which enables nodes to select the delivery rate using the information about the delivery path rather than some global information based on the interaction with other users is more appropriate. In order to design such a distributed rate configuration method for HTTP live video streaming, in this section we decompose **P2** into two sub-problems: a provider rate selection problem (PRSP) and an user rate adaptive problem (URAP), and consider both of them.

A. Provider Rate Selection Problem

By observing **P2**, we can easily find that providers can be separated according to eq. (6) yet coupled according to eq. (7). Hence, directly solving **P2** requires a centralized method coordinating all providers. First we consider the following equivalence problem of **P2**, which converts the constraints (7) into a set of linear combinations as follows:

P3:

$$\max_{\boldsymbol{x}\in[b_{\min},b_{\max}]^U} \quad \sum_{i\in\mathcal{S}}\sum_{i\in s_i(u)} J\left(x_{i,j}\right) \tag{9}$$

$$s.t. \quad \mathbf{X}_l\mathbf{1}_l^T \le \mathbf{1}_l^T.c_l, \quad l\in\mathcal{L} \tag{10}$$

where $|.|$ indicates the cardinality of set and $\mathbf{1}_l^T$ is a $|l(s)|$ dimension $\mathbf{1}$-vector whose all elements are 1. We denote matrix $\mathbf{X}_l = (x_{ji}^l)_{N_l\times|l(s)|}$ and $x_{ji}^l \in \{x_{j,i}|x_{j,i}\in s_j(u)_l, j\in l(s)\}$. Accordingly, each row of the $\mathbf{X}_l$ indicates a user rate combination of providers that use link l, and $\mathbf{X}_l$ lists all N_l possible combinations. The number of all possible combinations N_l corresponding to link l is equal to:

$$\prod_{i\in l(s)}\left|s_i(u)_l\right|$$

For instance, assume a network with 3 sources s_1, s_2 and s_3 over link l_1. Let $s_1(u)_{l_1} = \{x_{1,1}, x_{1,2}\}$, $s_2(u)_{l_1} = \{x_{2,1}\}$, $s_3(u)_{l_1} = \{x_{3,1}, x_{3,2}\}$. Thus, for the elements $x_{.1}, x_{.2}, x_{.3}$ of each row in $\mathbf{X}_{l_1}$, $x_{.1}\in\{x_{1,1},x_{1,2}\}$, $x_{.2}\in\{x_{2,1}\}$, $x_{.3}\in\{x_{3,1},x_{3,2}\}$. Therefore, $\mathbf{X}_{l_1}$ can be expressed as:

$$\begin{bmatrix} x_{1,1} & x_{2,1} & x_{3,1}\\ x_{1,2} & x_{2,1} & x_{3,1}\\ x_{1,1} & x_{2,1} & x_{3,2}\\ x_{1,2} & x_{2,1} & x_{3,2}\end{bmatrix}$$

Then, we can introduce the following proposition:

Proposition 1: Let $\boldsymbol{x}^*$ be the optimal value of **P2**; then exists a group of Lagrange multipliers $\boldsymbol{\lambda}^* = (\lambda_{11},\ldots,\lambda_{L\times N})$ and it can be shown that

(i)

$$\sum_{i\in\mathcal{S}}\sum_{j\in s(u)}\nabla_{\boldsymbol{x}}J\left(x_{i,j}^*\right) - \sum_{l\in\mathcal{L}}\sum_{j=1}^{N_l}\lambda_{lj}^*\left(\sum_{i=1}^{|l(s)|}\nabla_{\boldsymbol{x}}x_{ji}^{l\,*}\right) = 0 \tag{11}$$

(ii) for all $l\in\mathcal{L}$ and $j=\{1,\ldots,N_l\}$,

$$\lambda_{lj}^* \ge 0$$

(iii) for each link l, if $\lambda_{lj}^* > 0$, for all $i = 1,\ldots,|l(s)|$,

$$x_{ji}^{l\,*} = \max_{k\in[1,\ldots,N_l]} x_{ki}^{l\,*}$$

otherwise, $\lambda_{ji}^* = 0$.

Proof: Defining the Lagrangian $L(\boldsymbol{x},\boldsymbol{\lambda})$ of **P3**:

$$L(\boldsymbol{x},\boldsymbol{\lambda}) = \sum_{s\in\mathcal{S}}\sum_{i\in s(u)} J\left(x_{s,j}\right) - \sum_{l\in\mathcal{L}}\sum_{j=1}^{N_l}\lambda_{lj}\left(\sum_{i=1}^{|l(s)|}x_{ji}^l - c_l\right) \tag{12}$$

and corresponding Lagrange function of eq. (12):

$$D(\boldsymbol{\lambda}) = \sup_{\boldsymbol{x}}\left(\sum_{s\in\mathcal{S}}\sum_{j\in s(u)} J\left(x_{s,j}\right) - \sum_{l\in\mathcal{L}}\sum_{j=1}^{N}\lambda_{lj}\left(\sum_{i=1}^{|l(s)|}x_{ji}^l - c_l\right)\right) \tag{13}$$

the dual problem of **P3** is expressed as:

D1:

$$\begin{aligned} \min \quad & D(\boldsymbol{\lambda})\\ s.t. \quad & \lambda \ge 0 \end{aligned} \tag{14}$$

Because the primal problem **P3** is concave and constraints (10) satisfy the Slater condition [32], the optimal value of primal problem **P3** is equal to its dual **D1**. This means given the $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$ as the optimal solution of primal and dual problem, respectively, we have $f(\boldsymbol{x}^*) = D(\boldsymbol{\lambda}^*)$. Besides, as the constraints (9), (10) are continuous and differential and according to the Karush-Kuhn-Tucker condition [33], for the optimal rate configuration $\boldsymbol{x}^*$, there exists an unique Lagrange multiplier $\boldsymbol{\lambda}^* = (\lambda_{11}^*, \lambda_{12}^*\ldots,\lambda_{L\times N}^*)$, such that:

$$\begin{aligned}\nabla_{\boldsymbol{x}}L(\boldsymbol{x}^*,\boldsymbol{\lambda}^*) = &\sum_{s\in\mathcal{S}}\sum_{i\in s(u)}\nabla_{\boldsymbol{x}}J\left(x_{s,j}^*\right)\\ &-\sum_{l\in\mathcal{L}}\sum_{i=1}^{N}\lambda_{li}^*\nabla_{\boldsymbol{x}}\left(\sum_{i=1}^{|l(s)|}x_{ji}^{l\,*} - c_l\right)\\ &= 0\end{aligned} \tag{15}$$

In addition, recalling that $f(\boldsymbol{x}^*) = D(\boldsymbol{\lambda}^*)$, we have:

$$\lambda_{li}^*\left(\sum_{i=1}^{|l(s)|}x_{ji}^{l\,*} - c_l\right) = 0, \quad l\in\mathcal{L}, i = 1,2,\ldots,N$$

according to the complementary slackness [32], we have:

$$\begin{cases}\lambda_{li}^* > 0, \text{if}\left(\sum_{i=1}^{|l(s)|}x_{ji}^{l\,*} - c_l\right) = 0\\ \lambda_{li}^* = 0, \text{if}\left(\sum_{i=1}^{|l(s)|}x_{ji}^{l\,*} - c_l\right) < 0\end{cases} \tag{16}$$

Intuitively, for each link l, we have $\sum_{i=1}^{|l(s)|}\max_{i\in s_j(u)_l}x_{ji}^{l\,*} \ge \sum_{i=1}^{|l(s)|}x_{ji}^{*l}$. Namely, only when a linear combination $\sum_{i=1}^{|l(s)|}x_{ji}^{*l} = \sum_{i=1}^{|l(s)|}\max_{k\in[1,\ldots,N_l]}x_{ki}^{l\,*}$, equality $\sum_{j\in s(u)_l}x_{ji}^{l\,*} - c_l = 0$ may hold. Therefore, combining with (16), we can derive the following:

$$\begin{cases}\lambda_{li}^* \ge 0, \text{if}\sum_{i=1}^{|l(s)|}x_{ji}^{*l} = \sum_{i=1}^{|l(s)|}\max_{k\in[1,\ldots,N_l]}x_{ki}^{l\,*}\\ \lambda_{li}^* = 0, \text{if}\sum_{i=1}^{|l(s)|}x_{ji}^{*l} < \sum_{i=1}^{|l(s)|}\max_{k\in[1,\ldots,N_l]}x_{ki}^{l\,*}\end{cases} \tag{17}$$

hence, **P1** is proved. ■

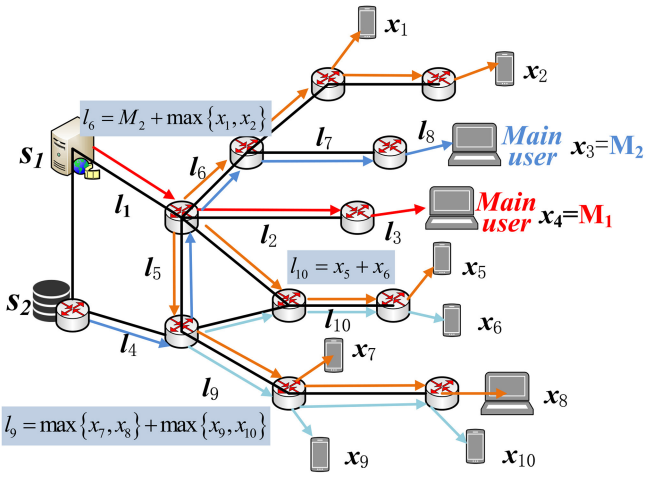


Fig. 2. An illustration of network with two provider s_1 and s_2 provider videos to users.

Now we will discuss how to enable each provider to determine the sending rate individually. Recall that $L(i)$ denotes the link set of provider i. We rewrite eq. (12) as follows:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_{i \in S} \left(\sum_{j \in s_i(u)} J(x_{i,j}) - \sum_{l \in L(i)} \sum_{k \in s_i(u)_l} \lambda_{i,l,k} x_{i,k} \right) + \sum_{l \in \mathcal{L}} \sum_{i=1}^{N_l} \lambda_{lj} c_l \tag{18}$$

Note that the first term of (18) is separable in terms of provider.

We define the user with $\max_{j \in s_i(u)} x_{s,j}$ as the *main user* and the other users are called *sub user*. Importantly, the sending rate M_i of provider i is equal to the rate of the *main user*. Therefore, for each provider i, there definitely exists a *main path* from provider i to its *main user* with link set $P_M(i)$, where

$$\max_{j \in s_i(u)_l} x_{i,j} = \max_{j \in s_i(u)} x_{i,j}, \forall l \in P_M(i)$$

We define the link belonging to $P_M(i)$ as the *main link* of i, and the other links in $L(i)$ as *sub links*. Figure 2 illustrates a scenario of two providers delivering two videos to a set of users. $\{l_1, l_2, l_3\}$ and $\{l_4, l_5, l_6, l_7, l_8\}$ are the *main path* of provider s_1 and s_2, respectively. The delivery rate over the *main path* of s_1, s_2 are equal to the main users x_3 and x_4. The lines with arrows colored in orange and cyan denote the flows to the *sub users* of s_1 and s_2, respectively. The delivery rates for providers over their *sub links* are equal to the maximum rates of the users that use these links, which may be less than the rate of the *main user*. For example, l_6 is the *main link* of s_1, and *sub link* of s_2. Hence, the total rate of link l_6 is $M_2 + \max\{x_1, x_2\}$. Based on the definition of *main path* and *main user*, eq. (18) can be further rephrased as:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_{i \in S} \left(J(M_i) + \sum_{j \in s_i(u)/\Lambda_i} J(x_{i,j}) - \sum_{l \in P_M(i)} \lambda_l M_i - \sum_{l \in L(i)} \sum_{k \in s_i(u)_l/\Lambda_i} \lambda_{ilk} x_{i,k} \right) + \sum_{l \in \mathcal{L}} \sum_{i=1}^{N} \lambda_{li} c_l \tag{19}$$

where Λ_i denotes the *main user* of provider i. Note, for $\forall i \in S$, $M_i^* \in \max\{x_{i,j}^* | j \in s_i(u)_l\}$. According to eq. (15) and eq. (11) in **Proposition 1**, we have $\frac{\partial L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)}{\partial M_i^*} = 0$, and therefore:

$$J'(M_i^*) = \sum_{l \in P_M(i)} \lambda_l^* \tag{20}$$

For each source i, $p_i = \sum_{l \in P_M(i)} \lambda_l^*$ and $M_i(p_i)$ denotes the delivery rate of provider as a function of p_i, according to eq. (20), the provider rate $M_i(p_i)$ is given by:

$$M_i(p_i) = J'^{-1}\left(\sum_{l \in P_M(i)} \lambda_l \right) \tag{21}$$

$\lambda_l^* (l \in P_M(i))$ of the dual problem $D(\boldsymbol{\lambda})$ can be derived by the gradient projection descend method [34] which iteratively approximates the optimal value $\boldsymbol{\lambda}^*$ along the gradient direction $\nabla D(\boldsymbol{\lambda})$. Specifically, for each link l, a sequence of $\{\lambda(t)\}_n$ is generated according to:

$$\lambda_l(t+1) = \lceil \lambda_l(t) - \gamma \frac{\partial D(\boldsymbol{\lambda})}{\partial \lambda_l} \rceil^+ \tag{22}$$

where the γ is the stepsize of each iteration. Since the $\nabla D(\boldsymbol{\lambda}^*) = 0$, the stop criterion $\lambda_l^* = \lambda_l(t)$ holds only when $\lambda_l(t) = \lambda_l(t-1)$.

As $\lambda_l^* \geq 0$, λ_l^* corresponding to the terms $\sum_{j=1}^{S} x_{ji}^{l\,*} r_j - c_l$ in eq. (12) is equal to zero, according to the **Proposition 1**. Namely, we only need to calculate λ_l corresponding to $\sum_{i \in l(s)} (\max_{j \in s(u)_l} x_{i,j} - c_l)$ in $D(\boldsymbol{\lambda})$. Therefore:

$$\frac{\partial D(\boldsymbol{\lambda})}{\partial \lambda_l} = c_l - \sum_{i \in l(s)} \max_{j \in s(u)_l} x_{i,j} \tag{23}$$

substituting eq. (23) into eq. (22), the descend rule of gradient projection for λ_l is as follows:

$$\lambda_l(t+1) = \left\lceil \lambda_l(t) - \gamma c_l - \sum_{i \in l(s)} \max_{j \in s(u)_l} x_{i,j}(t) \right\rceil^+ \tag{24}$$

Since c_l and for $\forall i \in l(s)$ x_i^l are local information for each link l, eq. (24) can be solved by each link locally and hence, a distributed algorithm can be applied. However, solving λ_l may require the rate of *sub users* since $\max_{j \in s(u)_l} x_{i,j}$ may not equal to M_i. In the next section, we will discuss the rate selection problem of *sub users*.

B. User Rate Adaptation Problem

In order to determine the rate of each user $x_{i,j}$, we decompose **P2** in terms of users and the corresponding sub-problem can be formulated as follows:

U1:

$$\max_{x_{i,j} \in [b_{\min}, b_{\max}]} \quad J(x_{i,j}) \tag{25}$$

$$s.t. \quad \sum_{k \in l(s)/i} \max_{j \in s_i(u)_l} x_{k,j} + x_{i,j} \leq c_l, \quad l \in p_j \tag{26}$$

$$x_{i,j} \leq \max_{j \in s_i(u)_l} x_{i,j}, \quad l \in p_j \tag{27}$$

where p_j indicates the link set used by user j. Constraint (26) indicates that for each link l used by j, the rate of j should

not exceed the minimum residual link capacity, and eq. (27) says that the playback rate of j cannot exceed the delivery rate of provider i over l. To illustrate the equivalence between **U1** and **P2**, we introduce the following theorem.

Theorem 2: For each user $j \in \mathcal{U}$, the corresponding optimal value x_j^* in **P2** can be derived equally by solving the problem **U1**. Namely, $\forall j$, x_j^* of **P2** and **U1** are equal.

Proof: By introducing the following parameters

$$x_i^l = \max\{x_{i,j} | j \in s_i(u)_l\}, \quad i = 1, \ldots, S; l = 1, \ldots, L$$

problem **P2** can be rephrased as follows:

P2:opt

$$\max_{\boldsymbol{x} \in [b_{\min}, b_{\max}]^U} \sum_{i \in \mathcal{S}} \sum_{i \in s(u)} J(x_{i,j}) \tag{28}$$

$$s.t. \sum_{i \in l(s)} x_i^l \le c_l \quad l \in \mathcal{L} \tag{29}$$

$$x_{i,j} \le x_i^l, \quad i \in \mathcal{S}, j \in s_i(u), l \in p_i, \tag{30}$$

Denote the optimal solution of **P2:opt** as $\boldsymbol{x}^*$. To derive the optimal value $\boldsymbol{x}^{*\prime}$ of **U1**, we aggregate **U1** of all users j and represent it as follows:

U1:A

$$\max_{x_{i,j} \in [b_{\min}, b_{\max}]} \sum_{i \in \mathcal{S}} \sum_{j \in s_i(u)} J(x_{i,j}) \tag{31}$$

$$s.t. \sum_{k \in l(s)/i} x_k^l + x_{i,j} \le c_l, l \in p_j, i \in \mathcal{S}, j \in \mathcal{U} \tag{32}$$

$$x_{i,j} \le x_i^l, \quad l \in p_j, i \in \mathcal{S}, j \in s_i(u) \tag{33}$$

Importantly, the theorem holds when $\boldsymbol{x}^* = \boldsymbol{x}^{*\prime}$. Now we prove $\boldsymbol{x}^* = \boldsymbol{x}^{*\prime}$. Given an utility function such as the one from eq. (1), eqs. (28)-(33) are differential and continuous, thus according to the Karush-Kuhn-Tucker condition, we have eqs.(53) and (54) shown at the bottom of p. 11. for **P2:opt** and **U1:A**, respectively.

According to **Proposition 1**,

$$\sum_{i \in \mathcal{S}} \sum_{j \in s(u)} \nabla_{\boldsymbol{x}} J\left(x_{i,j}^*\right) - \sum_{l \in \mathcal{L}} \lambda_l^* \nabla_{\boldsymbol{x}} \left(\sum_{i \in l(s)} x_i^{l*} - c_l \right) = 0 \tag{34}$$

Recall that **P3** is equivalent to **P2:opt**, so substituting eq. (34) into eq. (53), we have:

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

And because slackness complementary condition, there is:

$$\begin{cases} \upsilon_{ijl}^* > 0, x_{i,j}^* = x_i^{l*} \\ \upsilon_{ijl}^* = 0, x_{i,j}^* < x_i^{l*} \end{cases} \tag{35}$$

Using x^* to replace $x^{*\prime}$ in eq. (54), we have:

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

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

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

This contradicts with $x^{*\prime}$ being the maximum value.

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

Hence, in the above two cases, we have:

$$\sum_{i \in \mathcal{S}} \sum_{j \in s_i(u)} \sum_{l \in p_j} \lambda_{ijl}^{*\prime} \nabla_{\boldsymbol{x}} \left(\sum_{k \in l(s)/i} x_k^{l*} + x_{i,j}^* - c_l \right)$$

$$= \sum_{i \in \mathcal{S}} \sum_{j \in s_i(u)} \sum_{l \in p_j} \lambda_{ijl}^{*\prime} \nabla_{\boldsymbol{x}} \left(\sum_{k \in l(s)/i} x_k^{l*\prime} + x_{i,j}^{*\prime} - c_l \right) \tag{36}$$

As a result, eq. (54) is also equal to zero when replacing $x^{*\prime}$ with $x^{*\prime}$, and considering the minimum value of $U1{:}A$ is unique due to the concaveness, therefore, we have $x^{*\prime} = x^*$ and the theorem is proved. ■

To derive the optimal $x_{i,j}^*$ of **U1**, we consider the corresponding dual problems.

The Lagrangian of **U1** is:

$$L_u(x_{i,j}, \boldsymbol{\lambda}_{p_j}, \boldsymbol{\upsilon}_{p_j}) = J(x_{i,j}) - \sum_{l \in p_j} \upsilon_l \left(x_{i,j} - x_i^l \right)$$

$$- \sum_{l \in p_j} \lambda_l \left(\sum_{j \in l(s)/i} x_j^l + x_{i,j} - c_l \right) \tag{37}$$

The Lagrange dual function is:

$$D_u(\boldsymbol{\lambda}_{p_j}, \boldsymbol{\upsilon}_{p_j}) = \sup_{x_{i,j} \in [b_{min}, b_{\max}]} L_u(x_{i,j}, \boldsymbol{\lambda}_{p_j}, \boldsymbol{\upsilon}_{p_j}) \tag{38}$$

Then, the dual problem is:

U1:D:

$$\begin{aligned} \min \quad & D_u(\boldsymbol{\lambda}_{p_j}, \boldsymbol{\upsilon}_{p_j}) \\ s.t. \quad & \lambda_l \ge 0, \quad l \in p_j \\ & \upsilon_l \ge 0, \quad l \in p_j \end{aligned} \tag{39}$$

As **U1:A** is a concave optimization problem and satisfies the Slater condition [32], the strong duality holds. Namely, the optimal values of the primal and dual problems are equal. Thus, the primal optimal solution $x_{i,j}^*$ can be recovered from the dual optimal point $(\boldsymbol{\lambda}_{\boldsymbol{p_j}}^*, \boldsymbol{\upsilon}_{\boldsymbol{p_j}}^*)$:

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

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**, $x_{i,j}(p_j)$ can be derived as follows:

$$\frac{\mathrm{d}J\left(x_{i,j}(p_j)\right)}{\mathrm{d}x} = \sum_{l \in p_j} (\lambda_l + \upsilon_l) \tag{40}$$

$$\Rightarrow x_{i,j}(p_j) = J'^{-1} \left(\sum_{l \in p_j} (\lambda_l + \upsilon_l) \right) \tag{41}$$

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

$$\frac{\partial D_u}{\partial \lambda_l}\left(\boldsymbol{\lambda}_{p_j}, \boldsymbol{\upsilon}_{p_j}\right) = -\left(\sum_{k\in l(s)/i} x_k^l + x_{i,j} - c_l\right), l \in p_i \quad (42)$$

$$\frac{\partial D_u}{\partial \upsilon_l}\left(\boldsymbol{\lambda}_{p_j}, \boldsymbol{\upsilon}_{p_j}\right) = -\left(x_{i,j} - x_i^l\right), \quad l \in p_i \quad (43)$$

Therefore, based on eqs. (41), (42), (43), the dual problem **U1:D** can be solved by the following dual descend method, which is iteratively updated as follows:

$$x_{i,j}(t+1) \triangleq J'^{-1}\left(\sum_{l\in p_j}(\lambda_l(t) + \upsilon_l(t))\right) \quad (44)$$

$$\lambda_l(t+1) \triangleq \lambda_l(t) + \gamma\left(\sum_{k\in l(s)/i} x_k^l(t+1) + x_{i,j}(t+1) - c_l\right) \quad (45)$$

$$\upsilon_l(t+1) \triangleq \upsilon_l(t) + \gamma\left(x_{i,j}(t+1) - x_i^l(t+1)\right) \quad (46)$$

Because the flows of $l(s)$ and user j are delivered over link l, for each link l, x_i^l and $x_{i,j}$ can be obtained locally.

VII. Distributed Multi-source Rate Control Optimisation Algorithm (DMRCA)

Based on the already-presented decomposed sub-problems, in this section we introduce the distributed multi-source rate control optimisation algorithm (DMRCA) for HTTP live video streaming. We will also analyse the complexity, convergence and time adaptation of our proposed method.

A. Algorithm Design

The DMRCA design has two major parts: *processing at the user* and *processing at the provider*, which are described next.

Processing at the user: Recall that URAP can be solved by iteration of eqs. (44), (45), (46). Specifically, eq. (44) is separable in terms of users while eqs. (45), (46) can be processed locally at each link. Consequently, the process at the user side can be described as follows: in each iteration t, user solves the corresponding eq. (44) to derive $x_{i,j}(t)$ by collecting $\lambda_l(t-1)$ and $\upsilon_l(t-1)$ from links over its delivery path p_j, and communicates $x_{i,j}(t)$ to all the links over p_j. Link l receives the $x_{i,j}(t)$ of all users that use l and selects $\max_{j\in s_i(u)_l} x_{ij}(t)$ as $x_i^l(t)$ for each source s in $l(s)$. Then, link l uses x_i^l and $x_{i,j}$ to compute $\lambda_l(t+1)$ and $\upsilon_l(t+1)$ according to eqs. (45), (46). The derived $\lambda_l(t+1)$, $\upsilon_l(t+1)$ will be delivered to user j for computing the new $x_{i,j}$. The above-described process is repeated until the results reach the iteration criterion, $x_{i,j}(t+1) = x_{i,j}(t)$. The above process is fully distributed and does not require extra communication resources, since the information of $x_{i,j}(t)$,$\lambda_l(t)$,$\upsilon_l(t)$ is small enough and can be smuggled into data packets.

Processing at the provider: Similarly, at the provider side, each provider first determines the *main path* according to $\sum_{l\in p}\lambda_l$ of each path in the broadcast tree $s_i(l)$. As indicated by (21), the path with the minimum value of $\sum_{l\in p_j}\lambda_l$ will be set the *mainpath*. After determining M_i, provider i will send out a video with rate $\arg\min_{b_i\in\mathbb{B}}\|b_i - M_i\|$. Each link calculates λ_l according to eq. (24). As eq. (24) is equal to calculating eq. (45) with $\max_{j\in s_i(u)_l} x_{i,j}$ which is already computed as part of the processing at the user, (λ) can be derived directly by the following recursion process: let $lp(l,j)$ denote the link set between l to user j. The link l selects $\min_{j\in s_i(u)_l}\sum_{k\in lp(l,j)}\lambda_k$, aggregates it with its own λ_l and sends this $\min_{j\in s_i(u)_l}\sum_{k\in lp(l,j)}\lambda_k + \lambda_l$ to the upstream node. The upstreaming link repeats the above-described process until the provider is reached, when it stops.

The above processes at users and providers suggest treating users and node servers as processors in a distributed processing system, and the optimal rate of each user and provider can be derived only by communicating with links over the delivery path, without the need for coordination with other users or providers. This communication can be easily implemented by smuggling information into the data packets. Consequently, the proposed DMCRA is a fully distributed, lightweight and bit-rate optimized solution. The DMRCA pseudo-code is given in **Algorithm 1**.

B. Complexity Analysis

According to **Algorithm 1**, the complexity of links is mainly determined by the loop of the descend method and the number of providers and users that use each link. Let the descend method iterate N times, and the number of users and providers go through link l be U_l and S_l, respectively. Thus, the complexity of the algorithm from a link perspective is:

$$\log(N(U_l + S_l))$$

From an user and provider perspective, the corresponding complexity is mainly determined by the number of iterations of eqs. (44), (21), which are both N according to the descend method at the link. Therefore, the algorithm complexity is:

$$\log(N)$$

Based on this analysis, the complexity of the user and provider is dependent on the iteration, which is independent of the number of nodes. Thus, the algorithm has no scalability issue since the growing number of users will not significantly affect the efficiency of the algorithm.

C. Convergence Analysis

The proposed distributed DMRCA algorithm generates a sequence of $\{\boldsymbol{x}(t)\}$ approaching the optimal rate configuration $\boldsymbol{x}^*$. Naturally, there is the issue of whether the generated sequence converges to the optimal rate or not. Namely, for any $\varepsilon > 0$, there exists a T, such as we have:

$$\left\|\boldsymbol{x}(T) - \boldsymbol{x}^*\right\| \le \varepsilon$$

Next we discuss the condition of algorithm convergence.

Let $\widetilde{L} \triangleq \max_{p_j\in P}|p_j|$, where P is the set of all possible paths in the network and $\widetilde{S} = \max_{l\in\mathcal{L}}|l(s)|$. We have following theorem:

Algorithm 1: Distributed Rate Configuration Algorithm for HLS

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 use link l* **do**
    computes the $x_s^l(t)$ by $x_s^l(t) \leftarrow \max\{x_{i,j}(t)|j \in s(u)_l\}$;
  end
  foreach *user j go through the link l* **do**
    compute the $\lambda_l(t)$, $\upsilon_l(t)$ according to (45)(46);
    communicate the $\lambda_l(t)$, $\upsilon_l(t)$ with user j;
  end
  foreach *provider i use link l* **do**
    receive $\sum_{k\in lp(i,j)}\lambda_k$ from all down stream links in $s(l)$;
    $\sum_{k\in lp(l+1,j)}\lambda_k \leftarrow \max_{j\in s_i(u)_l}\sum_{k\in lp(l,j)}+\lambda_l$;
    send the $\sum_{k\in lp(l+1,j)}\lambda_k$ to upstream link;
  end
  t++;
end
$\boldsymbol{\lambda}^*=\boldsymbol{\lambda}(t)$,$\boldsymbol{\upsilon}^*=\boldsymbol{\upsilon}(t)$;
return $\boldsymbol{\lambda}^*$,$\boldsymbol{\upsilon}^*$;
user j's algorithm:
while $x_{i,j}(t)!=x_{i,j}(t+1)$ **do**
  receives the sum of $\sum_{l\in p_j}(\lambda_l(t)+\upsilon_l(t))$ from the links over its path;
  determines the next period delivery rate $x_{i,j}(t+1)$ by: $x_{i,j}(t+1) \triangleq J'^{-1}\left(\sum_{l\in p_j}(\lambda_l(t)+\upsilon_l(t))\right)$;
  communicates the $x_{i,j}(t+1)$ to links $l \in p_j$;
end
$x_j^*=x_{i,j}(t)$;
return x_j^*;
provider i's algorithm: **while** $M_i(t)!=M_i(t-1)$ **do**
  receives the sum of $\sum_{l\in P_M(i)}\lambda_l(t+1)$ from the broadcast tree; determines the new broadcasting rate $M_i(t+1)$ by $M_i(t+1)=J'^{-1}\left(\sum_{l\in P_M(i)}\lambda_l(t+1)\right)$;
  broadcast the video with rate $\arg\min_{b\in B_v}\|M_i(t+1)-b\|$;
end
return $M_i(t)$;
final;

Theorem 3: Suppose $J(x)$ is twice differential and for all $x \in [b_{\min}, b_{\max}]$, the corresponding $-J''(x_{i,j}) \geq \frac{1}{\widetilde{\alpha}_j}$, where $\widetilde{a} > 0$. Then, when step size $0<\gamma<\frac{1}{\widetilde{A}LS}$, where $\widetilde{A}=\max_{j\in\mathcal{U}}\alpha_j$ and from any initial point $\boldsymbol{x}(0)$, the $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\upsilon}^*)$ generated by **Algorithm 1** is dual optimal, namely, $\boldsymbol{x}^*$ is the optimal rate configuration for **P2**.

To prove this theorem with $J(x)$ formulated in eq. (1), we first introduce the following lemma.

Lemma 1: Given $J(x)$ as in eq. (1) over $[b_{\min}, b_{\max}]$, $J(x)$ is twice differential and the corresponding $J''(x)$ is bounded, namely, there exists a constant $\alpha>0$ such that $-J''(x)\geq\frac{1}{\alpha}$.

Proof: Starting from eq. (1), the corresponding twice differentiation is:

$$\begin{aligned}\left(-4.5e^{-0.77x}\right)'' &= -4.5\times -0.77^2 e^{-0.77x}\\ &< -4.5\times -0.77^2 e^{-0.77b_{\max}}\end{aligned} \tag{47}$$

Hence, $J''(x)$ is bounded and $\alpha = 4.5\times -0.77^2 e^{-0.77b_{\max}}$ ■

Next we give the proof for **Theorem 1**:

Proof: Let $\beta(j)=\frac{1}{-J''(x_i,j(p_j))}$, and let

$$A=(j)\begin{bmatrix}B(j) & 0\\ 0 & B(j)\end{bmatrix}=diag(\beta(j))_{2\mathcal{U}\times 2\mathcal{U}} \tag{48}$$

be a $2S\times 2S$ diagonal lumped matrix, where each $B(j)$ is $\mathcal{U}\times\mathcal{U}$ with diagonal elements $\beta(j), j\in\mathcal{U}$. According to eq. (40), we have:

$$\begin{aligned}J''(x_i,j(p_l))\frac{\partial x_{i,j}(p_{i,l})}{\partial p_{i,l}}&=1\\ p_{i,l}&=\begin{cases}\lambda_l, & i=1, l\in p_j\\ \upsilon_l, & i=2, l\in p_j\end{cases};\end{aligned} \tag{49}$$

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 eq. (48), we have following 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 eqs. (42), (43), 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: $\nabla^2(D_u(\boldsymbol{\lambda},\boldsymbol{\upsilon}))=CA(j)C$ According to the mean value theorem, $\forall \boldsymbol{m},\boldsymbol{n}$, we have:

$$\begin{aligned}\nabla D_u(\boldsymbol{m})-\nabla D_u(\boldsymbol{n})&=\nabla^2 D_u(\xi)(\boldsymbol{m}-\boldsymbol{n})\\ &=CA_j(\xi)C^T(\boldsymbol{m}-\boldsymbol{n})\end{aligned} \tag{50}$$

Based on the Schwartz inequality property of 2-norm $\|.\|$, further we have:

$$\|\nabla D_u(\boldsymbol{m})-\nabla D_u(\boldsymbol{n})\|\leq\left\|CA_j(\xi)C^T\right\|.\|\boldsymbol{m}-\boldsymbol{n}\|$$
$$\left\|CA_j(\xi)C^T\right\|^2\leq\left\|CA_j(\xi)C^T\right\|_\infty.\left\|CA_j(\xi)C^T\right\|_1$$

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,

$$\begin{aligned}\left\|CA_j(\xi)C^T\right\|_2&\leq\left\|CA_j(\xi)C^T\right\|_\infty\\ &=\max_i\sum_j\left[CA_j(\xi)C^T\right]_{i,j}\end{aligned}$$

$$
\begin{aligned}
&= \max_i \sum_j \sum_k \beta_k(w) R_{ik} R_{kj} \\
&= 2|p_j| \max_i \sum_k \beta_k(w) R_{ik} \\
&\le 2|p_j| \beta_k \max_i |l(i)| \\
&\le 2\widetilde{A}\widetilde{L}\widetilde{S} \qquad (51)
\end{aligned}
$$

Therefore, ∇D_u is Lipschitz with:

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

Thus, the sequence of $\{\boldsymbol{\lambda}(t), \boldsymbol{\upsilon}(t)\}$ generated by the gradient method is dual optimal. In addition, according to eq. (41), the primal optimal of $x^*_{i,j} = J'^{-1}(\sum_{l\in p_j}(\lambda^*_l + \upsilon^*_l))$. Because $J(.)$ is continuous and can be decoupled in terms of $x_{i,j}$, hence, $x_{i,j}(p_j)$ is continuous and therefore:

$$\lim_{t\to\infty} x_{i,j}(t) = x^*_{ij}$$

Namely, $x_{i,j}(t)$ converges to $x^*_{i,j}$, and the theorem is proved. ■

Another important issue is the convergence rate of the algorithm. In our algorithm, the optimal value is iteratively derived by the descend method. Let $p(t+1)$ be the sequence generated by the gradient descend method with $p(t+1) = p(t) - \gamma\nabla D_u$, and p^* be the optimal value. We then have:

$$
\begin{aligned}
&p(t+1) - p^* \\
&\quad = p(t) - p^* - \gamma\nabla D_u \\
&\quad = \int_0^1 1 - \gamma\nabla^2 D_u\big(x^* + \xi\big(p(t) - p^*\big)\big)\mathrm{d}\xi\big(p(t) - p^*\big) \qquad (52)
\end{aligned}
$$

from which we obtain by applying the 2-norm $\|.\|_2$:

$$
\begin{aligned}
&\big\|p(t+1) - p^*\big\|_2 \\
&\le \left(\left\|\int_0^1 1 - \gamma\nabla^2 D_u\big(x^* + \xi\big(p(t) - p^*\big)\big)\right\|_2 \mathrm{d}\xi\right)\big\|p(t) - p^*\big\|_2 \\
&\le \left\|1 - \gamma\nabla^2 D_u\big(x^* + \xi\big(p(t) - p^*\big)\big)\right\|_2 \big\|p(t) - p^*\big\|_2 \qquad (55)
\end{aligned}
$$

Thus, the convergence rate is bounded by:

$$\left\|1 - \gamma\nabla^2 D_u\big(x^* + \xi\big(p(t) - p^*\big)\big)\right\|_2$$

D. Time Varying Adaptation

Although in problem formulation, the objective function, video providers and routing are given and unchanged during the process, we can still directly extend our algorithm to an environment with time variable features such as dynamic caching and routing, and a time-dependent objective function. Importantly, the algorithm can still converge to the optimal solution when the network conditions change.

To cope with the time varying scenarios, the objective function **P2** can be re-formulated 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. $l(s)$ in constraint (7) is replaced by $l(s,t)$, which is the time variant provider set that use link l. Based on the above changes, each end users still executes the same user algorithm as in **Algorithm 1**, except for computing $p(j,t)$ in the place of (j) in eq. (44). Each link executes the same link algorithm as in **Algorithm 1** with the minor change of replacing $l(s)$ in eq. (45) with $l(s,t)$. Intuitively, if the change in link routing and providers is relative slower than the convergence rate of the algorithm we discussed, the algorithm still can converge to the optimal rates $\boldsymbol{x}^*$. We will illustrate this aspect in the experimental tests in Section VIII.

VIII. Performance Evaluation

The performance of HTTP live video streaming with DMRCA is evaluated using *ndnSIM 2.0* [35], a simulation tool based on Network Simulator 3 (NS-3). First, we present the simulation set-up in terms of network, video and user behaviors. Then, we describe the two scenarios considered. In the first scenario, we evaluate the bandwidth utilization and algorithm convergence at each link in a tree-based topology. The second scenario considers the American backbone topology, in which there are multiple sources and variable users. Users can obtain requested videos from multiple video providers, so the transmission path is also different. We compare our algorithm to the state-of-art solution HAVS-CCN and a traditional buffer-based adaptation method. HAVS-CCN optimizes the hop-by-hop content transmission in HTTP streaming. It directly adjusts video quality when DASH inaccurately estimating network throughput. We use the buffer-based approach provided by DASH as the traditional adaptation method. This algorithm raise the bit-rate when the buffer size reaches certain level. Our experiment measures the bandwidth utilization on different links in video transmission networks and the convergence value of the video bit rate during the video request process, which can be used to represent the user QoE.

A. Simulation Setup

In the simulation network, forwarding and content caching are the two main components, different forwarding and caching strategy may influence the performance significantly. Hence, we unify the forwarding and caching strategy that used in simulation. We select BestRoute as our forwarding strategy. In this strategy, each router maintains a routing table

$$\sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)} \nabla_{\boldsymbol{x}} J\left(x^*_{i,j}\right) - \sum_{l\in\mathcal{L}} \nabla_{\boldsymbol{x}}\lambda^*_l\left(\sum_{x\in l(s)} x^{l*}_i - c_l\right) - \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\nabla_{\boldsymbol{x}}\left(\upsilon^*_{ijl}x^*_{i,j} - x^{l*}_i\right) = 0 \qquad (53)$$

$$\sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)} \nabla_{\boldsymbol{x}} J\left(x^{*\prime}_{i,j}\right) - \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\lambda^{*\prime}_{ijl}\nabla_{\boldsymbol{x}}\left(\sum_{k\in l(s)/i} x^{l*\prime}_k + x^{*\prime}_{i,j} - c_l\right) - \sum_{i\in\mathcal{S}}\sum_{j\in s_i(u)}\sum_{l\in p_j}\nabla_{\boldsymbol{x}}\left(\upsilon^{*\prime}_{ijl}x^{*\prime}_{i,j} - x^{l*\prime}_i\right) = 0 \qquad (54)$$

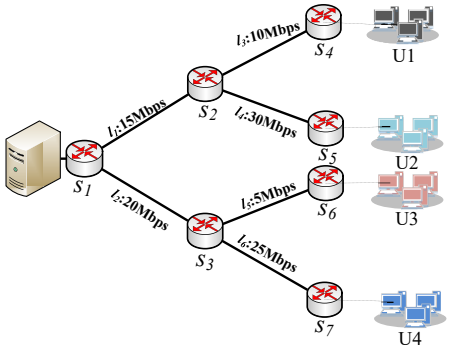


Fig. 3. Tree-based topology of scenario I.

Link Utilization Comparison		
	Case I	Case II
Link	[Mbps] Experim/ Optim/Capacity	[Mbps] Experim/ Optim/Capacity
l_1	14.9930/15/15	14.9010/15/15
l_2	20.0249/20/20	19.9154/20/20
l_3	10.0508/10/10	7.480/7.5/10
l_4	14.8748/15/30	7.4201/7.5/15
l_5	5.0041/5/5	4.991/5/5
l_6	20.0743/20/25	15.010/15/20

Fig. 4. Average link utilization comparison.

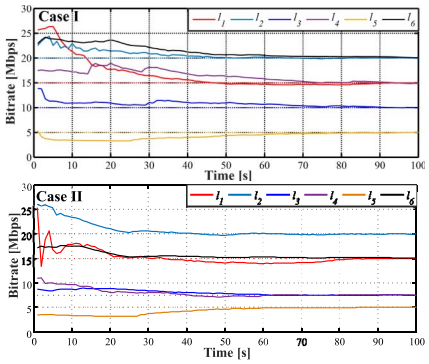


Fig. 5. Convergence rate at each link.

in order to support minimum hop counts content searching. For caching strategy, we employ the Leave Copy Everywhere (LCE), which enable the routers copying all passing content to CS and evicting them out when CS is full. For test video, we use MPEG-DASH multimedia streaming with SVC-encoded format. Each segment is two seconds long. And total test content catalogue contains 5 movies with 500s of each. The video delivered in the network can be provided using one base layer and four enhancement layers. The base layer b_1 has an average bit-rate of kbps, and enhancement layer 1, 2, 3, 4 have 600kps, 1600kps, 2600kps, 1940kps and 4440kps, respectively. Thus, there are 4 possible kinds of video representations, and their bit-rates are $b_1 = 600kps$, $b_1 + h_1 = 2200kps$, $b_1 + h_1 + h_2 = 4800kps$, $b_1 + h_1 + h_2 + h_3 = 6760kps$, $b_1 + h_1 + h_2 + h_3 + h_4 = 11200kps$. We set a group of users varies from 1 to 5 to start ask the same request at the same time. The arrival rate of users group follows the Poisson distribution with $\lambda = 0.1$. Each user group randomly select a video to request by a Zipf distribution. Specifically, the probability of requesting m-th popular video:

$$P(m) = \frac{m^{-\alpha}}{\sum_{k=1}^{M} k^{-\alpha}} \tag{56}$$

where α is the Zipf parameter with a value of 0.8, M denotes the total number of videos, which is 5 in our simulation. After determining the video to ask, end users will request the chunks of video in sequence and re-select a new video to request after requesting all chunks of the current video.

B. Experimental Results

Scenario 1 (Link Utilization Analysis): We focus on a tree-based network whose topology and link bandwidth are shown in Figure 3. In this topology, user groups $\{U1, U2, U3, U4\}$ only connect to the edge routers and each user group consists of 5 users. We consider two cases of user video requesting behaviors: *Case I*, all users start to request the video streaming at simulation starting time $t = 0$ and request the same video; *Case II*, each user group concurrently requests different videos at simulation start time $t = 0$.

Figure 4 shows the average link utilization in the two cases in comparison with the theoretical optimal value and actual link capacity. The theoretical value is computed by implementing our algorithm in MATLAB. As Figure 5 shows, for *Case I*, links l_1, l_2, l_3, and l_5 are the bottleneck links and thereby achieve full utilization. Because l_4 and l_5 are not bottleneck links, their bandwidth utilization are limited by the upstream links l_1 and l_2, respectively. For *Case II*, link l_1, l_2 and l_5 are still bottleneck links. However, due to the different videos each user group requested, the utilizations of l_3, l_4 and l_6 reduce to 7.5 Mbps, 7.5 Mbps and 15 Mbps, respectively. Figure 5 shows the comparison between theoretical and actual convergence rate at each link in both cases. As expected, in both cases, the proposed DMRCA algorithm also converges to the optimal values. However, it can be observed that DMRCA converges slower than theoretically. This is mainly because the iteration results exchange between users and links in realistic conditions experience a transmission delay. In addition, the iteration results are smuggled in *Interest* and data packets in our deployed algorithm, introducing an extra delay before sending. These delays slow down the convergence rate of the algorithm. In theoretical optimal computing, these delays are neglected and the convergence rate is only influenced by the iteration times and processing speed.

Scenario 2 (Performance Comparison): We consider the American backbone network topology as illustrated in Figure 6 for the performance comparison. In this network topology, each edge router builds links with 4 end users, and provides 1, 3, 5 and 10Mbps access bandwidth to each user, respectively. There are multiple video providers in this topology, which are illustrated as source 1 to source 3 in the figure. In this situation, a group of users are randomly selected to request the same video synchronously, while the request distribution of the user group follows the Zipf law as

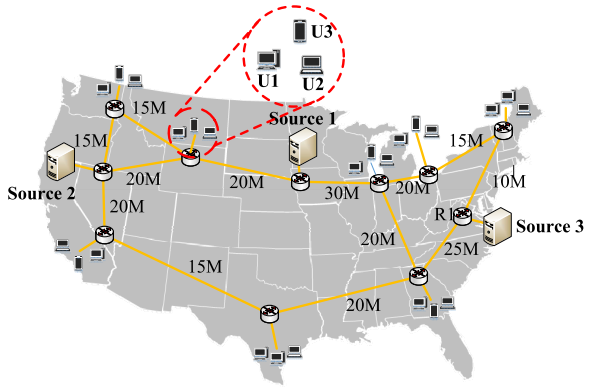


Fig. 6. Topology of American backbone network.

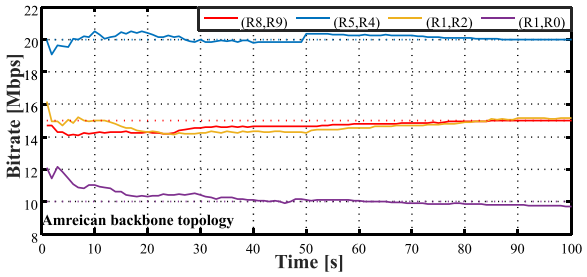


Fig. 7. Convergence analysis of links in American backbone topology.

in eq. (54) and the grouped requests arrive asynchronously according to a Poisson distribution.

We use various metrics to measure the performance of the algorithm, which are link utilization, bitrate and stalling. Link utilization reflects the actual utilization rate of network bandwidth under the regulation of algorithm, which determines the video transmission rate in the system, and further affects the following two metrics. Bitrate and stalling are the core factors affecting user experience. Bitrate determines the quality level of the video. Users can enjoy higher definition and smoother video content in the case of high bitrate. In the process of playing, if stalling occurs, it will have a direct negative impact on user experience. Therefore, the less stalling in the process of playing means the better user experience.

1) *Link utilization convergence analysis:* Figure 7 shows the link convergence of ($R8$,$R9$), ($R5$,$R4$), ($R1$,$R2$) and ($R1$,$R0$) in the American backbone topology. Note that in the topology figure, the links we select are bottleneck links and the total delivery rate of these links should equal to their link capacity. We note that the simulation behaved as expected, the delivery rate of each link converges to the theoretical optimal value, which is equal to the link capacity. In addition, from the convergence results, we also find that even when the network conditions vary (i.e., new users join in the network or caching on-path), the links still converge to the optimum value.

2) *User rate convergence analysis:* Figure 8 shows the rate convergence of users at router $R7$ in the American backbone topology. As the figure shows, the rate converges well to the theoretical results. Specifically, the variation of user rates can be explained as follows: user $U1$ first requests the video from $R6$. Because there are no other video flows, it can exclusively

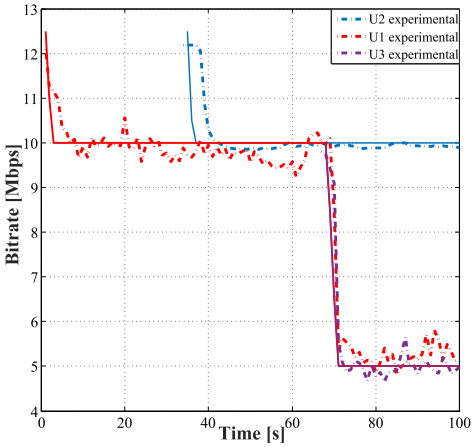


Fig. 8. Convergence analysis of U1-3 in American backbone topology.

use the link ($R6$, $R7$) achieving a maximum delivery rate of 10Mbps. When $U2$ joins the video distribution system, it accesses the video from ($R7$, $R8$) and since there is a near copy of the asked content at $R8$, it does not influence the rate of $U1$. At 66s, when other flows from $U3$ pass over the link ($R6$, $R7$), the link bandwidth of ($R6$, $R7$) is used by two flows simultaneously and the rate of $U1$ decreases to 5Mps.

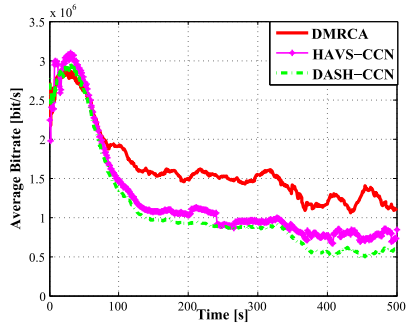
3) *Average bit rate (ABR) comparison:* we define the ABR as the arithmetic mean of average bit-rate of overall users, we calculate the ABR at time T by:

$$ABR(T) = \frac{1}{UT}\sum_{u=1}^{U}\sum_{t=1}^{T} BR_u(t) \tag{57}$$

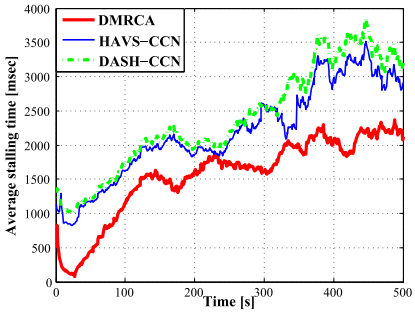
where U is the total number of users and $BR_u(u)$ indicates the bit-rate of user u at time t.

Figure 9(a) depicts the ABR in the American backbone topology. As the figure shows, the ABRs of the three solutions compared experience an increasing trend at the beginning. After 200s, all solutions decrease their rates and then enter a periodical vibration phase. This phenomenon can be explained as follows. Initially, the network load is low, allowing the link to accommodate all high bitrate video requests. However, as the number of end users increases, the link's capacity restricts the growth of the average user throughput, leading to a decrease in ABR. In the latter half of the simulation, frequent user joining and leaving activities causes ABR to dynamically fluctuate with the number of active users.

The results of Figure 9(a) suggest an increase of 30% and 41% of ABR in favor of DMRCA when compared with the other two solutions. In DMRCA, the overall bitrate is distributively optimized and converges to the theoretical optimal, hence, providing the best performance among the three solutions. HAVS-CCN adjusts the data rate at each hop locally, and fails to optimize the overall user bitrate. Each client in DASH-CCN greedily requests higher bitrate videos in order to maximize their own bitrate, which aggravates the network congestion when the network is already in a high load condition. Therefore, DASH-CCN has the worst performance in terms of ABR.



(a) ABR of American backbone topology



(b) PFF of American backbone topology

Fig. 9. Average cache hit ratio vs. simulation time along 2 sizes of video sets: (a) $|V| = 30$; (b) $|V| = 40$.

4) *Playback freeze frequency (PFF) comparison:* We define PFF as the average occurrence of freeze per second during the simulation. The lower PFF is, the smoother playback experienced by the client is. Figure 9(b) shows PFF in tests with the American backbone topology. The results show that when DMRCA is employed, PFF decreases with about 20% and 25% in comparison with the values experience by the other two solutions. As mentioned, DMRCA uses a distributed optimization method in order to fully use the link bandwidth while also avoiding the network congestion by limiting the total delivery rate to the link capacity and hence achieves a smoother playback. HAVS-CCN also limits the data rate to the link capacity at each hop, hence avoiding the network congestion. DASH-CCN using a greedy method to request video content with a high risk of playback freeze when the available bandwidth is not enough to support smooth playback of high bitrate videos.

C. Discussion

DMRCA is a distributed rate control algorithm, so it needs to be deployed at every node in the network. In general this is associated with a large system deployment cost. Therefore, for simple network architectures with fewer distributed nodes, the optimization introduced by a possible deployment of DMRCA is limited, considering the deployment cost. However, for large deployments, the benefit of employing DMRCA is significant. Therefore, an interesting research avenue is to explore for what range of network topologies DMRCA is most suitable, and consider the deployment decision from both deployment cost and bandwidth utilization optimization points of view.

In addition, DMRCA is designed to select bitrate solely based on information received from the connected link. Therefore, it is obviously optimized for certain use cases, but it is not necessary for all use cases. If there is a central server in the network structure that provides information about network links such as congestion status and available bandwidth to all network nodes in a low-cost manner, the DMRCA calculation process can actually be replaced by this mechanism, because the core focus of DMRCA is to infer the optimal bit rate selection from link usage. In a centralized structure, the rate selection of each node can be uniformly performed by the central node.

IX. Conclusion and Future Work

This study introduced an innovative distributed multi-source optimal bitrate control algorithm (DMRCA) for adaptive video streaming. It includes a formulation of the rate control problem as a concave MAP, which was decomposed into two sub-problems, PRSP and URAP. Following a demonstration of the equivalence between the original problem and the two sub-problems, DMRCA was proposed as a distributed optimal solution that enables users and providers to communicate through links and achieve optimal rate control. The paper discussed the complexity, convergence, and time-varying adaptability of the proposed algorithm. Simulation results demonstrated the superiority of DMRCA over other state-of-art solutions. Future work will involve designing an online asynchronous algorithm to facilitate deployment in highly-dynamic mobile environments.

References

[1] L. R. Solano-Hurtado and M. M. Soto-Cordova, “A study on video streaming quality of DASH scheme in multimedia services,” in *Proc. IEEE CONIITI*, 2021, pp. 1–4.

[2] S. Duraimurugan and P. J. Jayarin, “Analysis and study of multimedia streaming and congestion evading algorithms in heterogeneous network environment,” in *Proc. IEEE 2nd Int. Conf. Intell. Comput. Control Syst.*, 2018, pp. 1248–1252.

[3] D. Wu, J. Yang, H. Wang, B. Yang, and R. Wang, “Terminal-edge-cloud collaboration: An enabling technology for robust multimedia streaming,” in *Proc. IEEE 16th Int. Conf. Mob., Sens. Netw.*, 2020, pp. 427–434.

[4] D. Samiayya, J. Ramasamy, and M. Gunasekar, “An efficient congestion control in multimedia streaming using adaptive BRR and fuzzy butterfly optimization,” *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 3, 2023, Art. no. e4707.

[5] *Global Video Streaming Market 2023-2027*. 2022. [Online]. Available: https://www.researchandmarkets.com/reports/5513566/global-video-streaming-market-2023-2027

[6] M. J. Alam, M. R. Hossain, S. Azad, and R. Chugh, “An overview of LTE/LTE-A heterogeneous networks for 5G and beyond,” *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 8, 2023, Art. no. e4806.

[7] A. Polakovic, G. Rozinaj, and G.-M. Muntean, “User gaze-driven adaptation of omnidirectional video delivery using spatial tiling and scalable video encoding,” *IEEE Trans. Broadcast.*, vol. 68, no. 3, pp. 609–619, Sep. 2022.

[8] 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.

[9] 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.

[10] A.-N. Moldovan and C. H. Muntean, “DQAMLearn: Device and QoE-aware adaptive multimedia mobile learning framework,” *IEEE Trans. Broadcast.*, vol. 67, no. 1, pp. 185–200, Mar. 2021.

[11] A. Yaqoob, C. H. Muntean, and G.-M. Muntean, "Flexible tiles in adaptive viewing window: Enabling bandwidth-efficient and quality-oriented 360°VR video streaming," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, 2022, pp. 1–8.

[12] Z. Ye et al., "VRCT: A viewport reconstruction-based 360° video caching solution for tile-adaptive streaming," *IEEE Trans. Broadcast.*, vol. 69, no. 3, pp. 691–703, Sep. 2023.

[13] L. Zhong, M. Wang, C. Xu, S. Yang, and G.-M. Muntean, "Decentralized optimization for multicast adaptive video streaming in edge cache-assisted networks," *IEEE Trans. Broadcast.*, vol. 69, no. 3, pp. 812–822, Sep. 2023.

[14] A. Yaqoob and G.-M. Muntean, "Fuzzy logic-based adaptive multimedia streaming for Internet of Vehicles," in *Proc. IEEE 97th Veh. Technol. Conf.*, Florence, Italy, 2023, pp. 1–6.

[15] G. Zhou, Z. Luo, M. Hu, and D. Wu, "PreSR: Neural-enhanced adaptive streaming of VBR-encoded videos with selective prefetching," *IEEE Trans. Broadcast.*, vol. 69, no. 1, pp. 49–61, Mar. 2023.

[16] T. Lamsub and P. Tandayya, "Dynamic popularity caching policy for dynamic adaptive streaming over HTTP," in *Proc. IEEE 19th Int. Symp. Commun. Inf. Technol. (ISCIT)*, 2019, pp. 322–327.

[17] S. Mori, Y. Mizoguchi, and M. Bandai, "An HTTP adaptive streaming method considering motion intensity," in *Proc. IEEE 7th Int. Conf. Cloud Netw. (CloudNet)*, 2018, pp. 1–3.

[18] D. Liu, X. Tan, Y. Liu, Y. He, and Q. Zheng, "Reinforcement learning based dynamic adaptive video streaming for multi-client over NDN," in *Proc. IEEE 4th Int. Conf. Hot Inf.-Centric Netw. (HotICN)*, 2021, pp. 68–73.

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

[20] H. Yuan, H. Lu, L. Meng, and M. Liu, "MUABR: Multi-user adaptive bit-rate algorithm based multi-agent deep reinforcement learning," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 751–756.

[21] A. O. El Meligy, M. S. Hassan, and T. Landolsi, "A buffer-based rate adaptation approach for video streaming over HTTP," in *Proc. IEEE Wireless Telecommun. Symp. (WTS)*, 2020, pp. 1–5.

[22] A.-T. Tran, N.-N. Dao, and S. Cho, "Bit-rate adaptation for video streaming services in edge caching systems," *IEEE Access*, vol. 8, pp. 135844–135852, 2020.

[23] H. Phan, D. Nguyen, H. T. T. Tran, T. Thu Huong, and T. C. Thang, "Application layer throughput control for video streaming over HTTP2," in *Proc. IEEE 8th Int. Conf. Commun. Electron. (ICCE)*, 2021, pp. 123–128.

[24] E. Volodina and E. P. Rathgeb, "Flow control in the context of the multiplexed transport protocol QUIC," in *Proc. IEEE 45th Conf. Local Comput. Netw. (LCN)*, 2020, pp. 473–478.

[25] T. Dai, X. Zhang, Y. Zhang, and Z. Guo, "Statistical learning based congestion control for real-time video communication," *IEEE Trans. Multimedia*, vol. 22, no. 10, pp. 2672–2683, Oct. 2020.

[26] T. Mai, H. Yao, X. Zhang, Z. Xiong, and D. Niyato, "A distributed reinforcement learning approach to in-network congestion control," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, 2020, pp. 817–822.

[27] J. Lin, H. Lin, Y. Xu, Y. Kang, and T. Zhao, "Virtual-competitors-based rate control for 360-degree video coding," *IEEE Trans. Broadcast.*, vol. 70, no. 1, pp. 357–365, Mar. 2024.

[28] M. Zhou, X. Wei, C. Ji, T. Xiang, and B. Fang, "Optimum quality control algorithm for versatile video coding," *IEEE Trans. Broadcast.*, vol. 68, no. 3, pp. 582–593, Sep. 2022.

[29] Y. I. Choi and C. G. Kang, "Scalable video coding-based MIMO broadcasting system with optimal power control," *IEEE Trans. Broadcast.*, vol. 63, no. 2, pp. 350–360, Jun. 2017.

[30] C. Concolato et al., "Adaptive streaming of HEVC tiled videos using MPEG-DASH," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1981–1992, Aug. 2018.

[31] A. Seufert, F. Wamser, D. Yarish, H. Macdonald, and T. Hoßfeld, "QoE models in the wild: Comparing video QoE models using a crowdsourced data set," in *Proc. 13th Int. Conf. Qual. Multimedia Exp. (QoMEX)*, 2021, pp. 55–60.

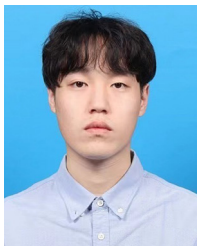
[32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ., 2004.

[33] D. P. Betsekas, *Nonlinear Programming*, 2nd ed. Nashua, NH, USA: Athena Sci., 1999.

[34] D. P. Betsekas and J. N. Tsitsiklis, *Parallel and Distributed Compuation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

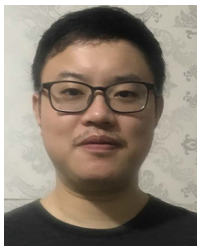
[35] "ndnSim in network simulator 3 (NS-3)." Accessed: Oct. 2023. [Online]. Available: http://ndnsim.net/intro.html

Shujie Yang received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017, where he is an Associate Professor. His current research interests include the areas of VR networks, content delivery network, and wireless networking.



Chuxing Fang received the B.S. degree in e-commerce and law from the Beijing University of Posts and Telecommunications, Beijing, China in 2022, where he is currently pursuing the master's degree with the School of Computer Science. His research interests include multimedia transmission and virtual reality.

Lujie Zhong received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2013. She is currently a Professor with the Information Engineering College, Capital Normal University, Beijing. She has published papers in prestigious international journals and conferences in related areas, 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 received the Ph.D. degree in computer technology from the Beijing University of Posts and Telecommunications, China, in 2020, where he currently serves as an Associate Researcher with the State Key Laboratory of Network and Switching Technology. His research interests include information centric networking, wireless communications, and multimedia sharing over wireless networks.

Zan Zhou (Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2022, where he is currently a Postdoctoral Fellow with the School of Computer Science. His research interests include network security, artificial intelligence privacy, and active defense.

Han Xiao (Member, IEEE) received the Ph.D. degree in computer science and technology from the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China, in 2022, where he is currently a Postdoctoral Fellow with the State Key Laboratory of Networking and Switching Technology. His research interests include immersive media transmission, online learning, and resource management.

Hao Hao received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China,in 2021. He is currently a Lecturer with the Shandong Computer Science Center (National Supercomputing Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences). His research interests include MEC, and content caching over the wireless network and multimedia communications.

Changqiao Xu (Senior Member, IEEE) is a Professor with the School of Computer Science and Technology and the Deputy Director of the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China. He has 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 mobile networking, multimedia communications, and future Internet technology. He is currently serving as the Editor-in-Chief of *Transactions on Emerging Telecommunications Technologies* (Wiley). He has served a number of international conferences and workshops as the co-chair and a TPC member.

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. He coordinated the EU Project NEWTON and leads the DCU Team in the EU Projects TRACTION and HEAT. 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.