A Multi-user Cost-efficient Crowd-assisted VR Content Delivery Solution in 5G-and-beyond Heterogeneous Networks

Lujie Zhong, Xingyan Chen, Changqiao Xu, *Senior Member, IEEE,* Yunxiao Ma, Mu Wang, Yu Zhao, and Gabriel-Miro Muntean, *Senior Member, IEEE,*

Abstract—The latest evolution of wireless communications enables users to access rich Virtual Reality (VR) services via the Internet, including while on the move. However, providing a premium immersive experience for the massive number of concurrent users with various device configurations is a significant challenge due to the ultra-high data rate and ultra-low delay requirements of VR livecast services. This paper introduces an innovative multi-user cost-efficient crowd-assisted delivery and computing (MEC-DC) framework, which leverages mobile edge computing and end-user resources to support high performance VR content delivery over 5G-and-beyond heterogeneous networks (5G-HetNets). The proposed MEC-DC framework is based on three main solutions. First is a novel buffer-nadir-based multicast (BNM) mechanism for VR transmissions over 5G-HetNets. BNM ensures smooth and synchronized user viewing experiences by maximizing the average playback buffer-nadir of all participants with stochastic optimization. Second and third are practical distributed algorithms: the cost-efficient multicast-aware transcoding offloading (MATO) and crowd-assisted delivery algorithm (CAD) which optimize jointly multicast delivery and video transcoding. The algorithms' optimality and complexity were investigated. The proposed MATO-CAD solution was evaluated with real datasets, trace-driven numerical simulations, and prototype-based experiments. The trace-driven experimental results showed how the proposed solution provides 18% throughput improvement, the lowest delay, and the best playback freeze ratio in comparison with three other state-of-the-art solutions.

Index Terms—Virtual reality, 5G-and-beyond heterogeneous network, content delivery, multicast, video transcoding

1 Introduction

The latest innovations in mobile communication technologies, including the advancements related to the fifth-generation (5G)-and-beyond networks provide support for ultra-high data rate, ultra-low latency, ubiquitous access, and highly mobile computing [1, 2]. These offer a solid foundation for advanced live streaming services such as panoramic video and virtual reality (VR) [3–6]. Recently, Facebook, one of the world's largest social network technology companies, changed its name to Meta[1] (i.e., metaverse) to describe the vision of future human work and life with VR services. This move gave a further boost to the world-wide VR market, which is expected to reach $454.73 billion by 2030 according to recent market research[2]. However, it is foreseeable that millions of concurrent accesses by global users with different device configurations make it significantly challenging to provide high-quality immersive VR services, especially in a wireless environment, despite the latest 5G support [7, 8]. On one hand, according to a Huawei Cloud VR white paper[3], providing an ideal strong-interaction VR experience for a single user requires ultra-high data rate (i.e., 1GMbps bandwidth) and ultra-low delay (i.e., less than 8ms latency). On the other hand, in order to target high quality of experience (QoE), service providers need to adapt the VR content delivery to various devices and network configurations. The latest solutions employ online transcoding for live VR services by dividing the video into tiles, encoding the tiles into multiple resolutions, and selecting the ones appropriate for the user's viewing environment [9–13]. Since each user has a personalized region of interest, the multi-resolution tiles are stitched, and a specific panoramic video is formed (sometimes with a high-resolution viewport and a low-resolution background) for individualized services. As a consequence, VR livecast services are associated with very demanding real-time computing capabilities and have large bandwidth needs [12, 13].

Cloud computing has become a natural choice to meet the computation-intensive requirements of online transcod-

- *This work was supported in part by the National Natural Science Foundation of China under Grant 61872253, 61871048. G.-M. Muntean acknowledges the Science Foundation Ireland Research Centres Programme support under Grant Nos. SFI/12/RC/2289 P2 (Insight Centre for Data Analytics) and 16/SP/3804 (ENABLE). The corresponding author is Xingyan Chen.*
- *L. Zhong is with Information Engineering College, Capital Normal University, Beijing 100048, P. R. China. E-mail: zhonglj@cnu.edu.cn.*
- *X. Chen and Y. Zhao are with Financial Intelligence and Financial Engineering Key Laboratory of Sichuan Province, School of Economic Information Engineering, Southwestern University of Finance and Economics, SWUFE 611130, P. R. China. E-mail: {xychen, zhaoyu}@swufe.edu.cn.*
- *C. Xu and Y. Ma are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, P. R. China. E-mail: {cqxu, myx}@bupt.edu.cn.*
- *M. Wang is with Department of Computer Science and Technology & BNRist, Tsinghua University, Beijing 100084, P. R. China. E-mail: wangmu@bupt.edu.cn.*
- *G.-M. Muntean is School of Electronic Engineering, Dublin City University, Glasnevin, Dublin 9, Ireland, E-mail: gabriel.muntean@dcu.ie*

[1] https://www.cnn.com/2021/10/28/tech/facebook-mark-zuckerberg-keynote-announcements/index.html

[2] https://www.alliedmarketresearch.com/augmented-and-virtual-reality-market

[3] https://www.huawei.com/minisite/pdf/ilab/cloud_vr_network_solution_white_paper_en.pdf

ing of live VR [14–18]. For example, Chen *et al.* in [14] focused on computing resource management in wireless networks and proposed a novel Cloud-enabled resource allocation framework to enhance user immersive experience. However, due to the geographically remote distance between Cloud and user, VR services are prone to high transmission delays and congestion in the resource-limited mobile network [7–10]. In this context, offloading the intensive tasks to the edge has attracted extensive attention in recent years as a promising technique to overcome the above challenges [4, 5, 7–10, 19–28]. Fig. 1 shows a typical situation involving an edge-assisted VR livecast system over a 5G HetNet. Dai *et al.* in [19] proposed a viewport-based VR caching system over Cloud Radio Access Network (C-RAN) to facilitate the view synthesis and content allocation using mobile edge computing (MEC) and hierarchical caching technologies. Additionally, many other edge-assisted solutions have been proposed to improve the efficiency of computing resource allocation, including some which employ deep reinforcement learning [21], distributed content rendering [22], and scalable multi-layer VR video tiling [24].

Once the VR video content is prepared, it needs to be delivered efficiently to users, as wireless networks' bandwidth resources are limited. Due to the broadcast nature of both live streaming concept and wireless network functionality, multicast is a promising transmission technology that can utilize the available bandwidth resources efficiently by aggregating viewers' requests for the same content and servicing them in a single session. Reusing content for multiple users with overlapping FoVs when employing multicast delivery is another critical issue considered in the research literature [11, 29–34]. For example, Perfecto *et al.* in [11] proposed an online deep learning-based multicast solution for rate-adaptive streaming by leveraging deep recurrent neural networks. To improve resource efficiency, Dang *et al.* in [30] and Sun *et al.* in [31] considered jointly communications, caching, and computing resource allocation to reduce the system cost and improve service quality. These studies also modeled this joint optimization problem and revealed communications-caching-computing trade-offs.

In our previous works [33, 34], an augmented queue-based structure [33] was built over a Cloud-Edge-Crowd integrated infrastructure to jointly optimize data transmission and online transcoding for livecast services. Then, an augmented graph model [34] was employed to transform the joint allocation optimization of computing resources and transmission resources into a generalized network routing problem. We have designed a distributed actor-critic algorithm to solve the above problem by finding a low-latency, high-efficient transcode delivery path for each user. However, most of the research has focused on alleviating the system overhead and reducing the transmission latency and has ignored the immersive experience performance for multiple users. High-quality user experience also requires minimizing the time-shift between multiple users to ensure an immersive viewing experience for every user in the virtual world. Specifically, providing premium immersive experiences for geo-distributed users with different devices and network configurations over resource-limited heterogeneous networks is significant, although very challenging.

For clarity, the following are this work's main challenges:

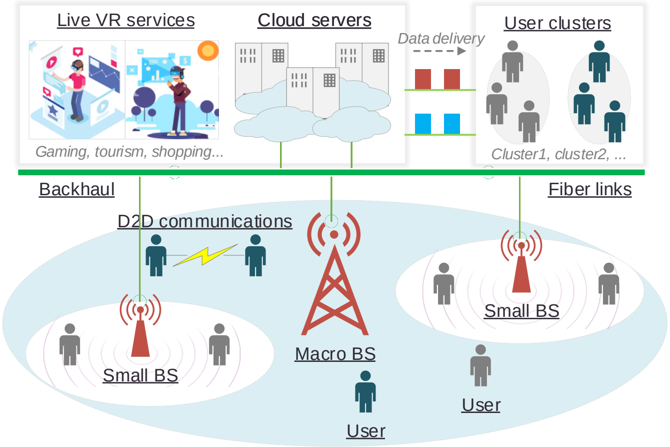


Fig. 1. Illustration of an edge-assisted livecast VR system in a 5G HetNet including 1) a remote server that provides varieties of VR content, such as VR cloud games; 2) users divided into multiple multicast clusters based on the accessed content 3) a 5G HetNet includes one macro BS and a series of small BSs with edge servers. 4) device-to-device communication support, which enables users to share content directly.

1) The delivery system needs to adjust frequently the delivery and processing task allocation between different base stations (BSs) and user devices to provide cost-efficient VR live streaming while adapting to configurations of user devices and network conditions.
2) Ultra-dense 5G HetNets with device-to-device communication support provide multiple alternative access points for users, creating opportunities for users to retrieve content via multiple transmission paths. However, this complicates the process of problem-solving and makes optimal resources control more difficult.
3) Since the system needs to transcode and deliver the VR video content to a large number of users, designing a joint optimization solution that efficiently utilizes the distributed computing resources while providing high-quality VR services is non-trivial.
4) The time-shift issue makes providing an immersive VR experience for multiple participants significantly even more challenging as users have both different configurations of networks and devices and various latencies.

Motivated by these challenges, an innovative **Multi-user Cost-efficient Crowd-assisted Delivery and Computing (MEC-DC)** framework is introduced by leveraging mobile edge computing to support achieving VR video processing flexibility and content delivery efficiency. The framework is based on a novel solution that consists of three main innovations. First, a novel **Buffer-nadir-based Multicast (BNM) mechanism** for multicast scheduling in 5G HetNets is proposed. BNM is based on a new concept denoted *buffer-nadir* and employs the *age of information (AoI)* idea introduced by Modiano *et al.* [35–37]. AoI quantifies the freshness of the receiver's knowledge about the sender. By jointly considering each user's playback and buffer level, the multicast scheduling is modeled as a stochastic process of the buffer charging problem. Second, **an approach for Cost-efficient Multicast-aware Transcoding Offloading (MATO)**, which adjusts the transcoding task allocation among edge servers based on a multicast decision is introduced. MATO considers the broadcast nature of live streaming services and employs computing resources provided by edge servers in

5G-HetNets to facilitate VR video transcoding. Third, a practical online **Crowd-assisted Delivery algorithm (CAD)** was described to facilitate VR video delivery by using device-to-device (D2D) communications.

The contributions of this paper are as follows:

1) We propose the MEC-DC framework and provide a novel buffer evolution model that introduces a new concept of *buffer-nadir* to address the time-shift issue and capture the quality of multi-users immersive experience.
2) To support an immersive experience, we formalize the multicast problem over 5G-HetNets as a stochastic buffer charging problem which maximizes the average buffer level for all viewers fairly.
3) We present an approximately optimal solution of multicast and task offloading. We describe the two distributed algorithms MATO and CAD, which support cost-efficient and high-quality VR livecast services.
4) We prove that our MATO-CAD solution finds a nearly optimum, up to an additive factor (half the number of user clusters) away from the optimal buffer size. To the best of our knowledge, this paper opens new avenues for improved quality live VR services by performing innovative joint computing and transmission resource allocation in 5G-HetNets.
5) We evaluate the MATO-CAD solution in terms of theoretical performance and conduct several trace-driven simulations. We compare MATO-CAD with three state-of-the-art methods [7, 31, 38]. Results show that our method outperforms the others in terms of throughput, latency, resource cost, and quality of experience (QoE).

This paper is organized as follows. Section II discusses related works. Section III introduces the system model. Section IV formalizes the problem and section V designs two practical algorithms. Section VI includes simulation results and section VII draws conclusions and future work.

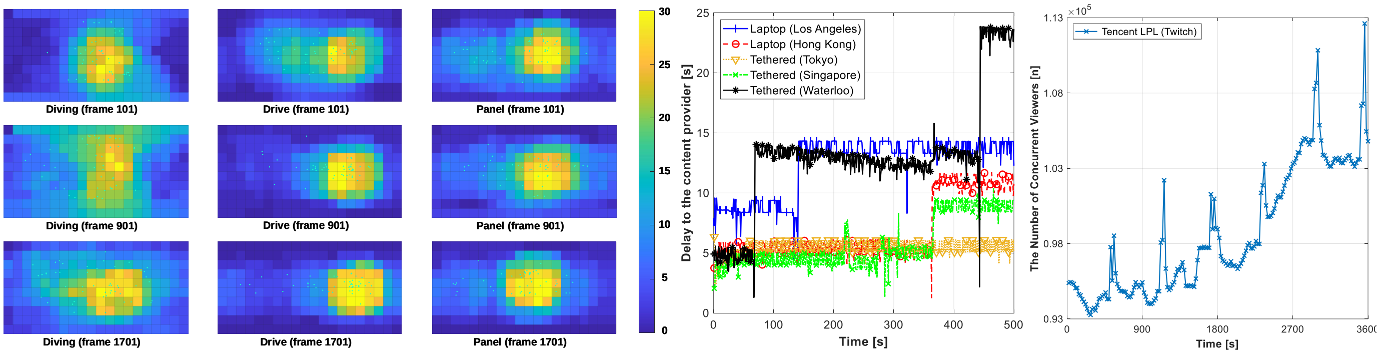
2 Background

Many mobile computing solutions have already been proposed for live VR. However, very few studies consider joint multicast and transcoding optimization for content delivery over 5G HetNets based on device-to-device communications. This research faces complex challenges when trying to provide a high-quality immersive VR experience while saving system resources. To the best of our knowledge, this paper presents the first attempt to employ AoI technology assisting joint computing and transmission resource allocation over 5G HetNets. Next, several works related to mobile edge computing and multicast for live VR are discussed.

2.1 Live VR Streaming With Mobile Edge Computing

Live VR video transcoding solutions mainly fall into two categories: centralized cloud approaches [14–18] and distributed computing solutions [4, 5, 7, 8, 10, 19–28]. In centralized solutions, VR video processing is mainly performed on dedicated cloud servers, which can provide stable computing resources. For example, Simiscuka *et al.* [15] considered a novel social VR-IoT scene and proposed a cloud-based solution to provide computing resources for multiple geo-distributed users. However, the proposed method finds approximate solutions, leading to a degradation in user experience quality. To provide a highly immersive experience for users, the authors of [16] proposed a cloud-based iterative semi-Lagrangian method to simulate the interaction in a VR scene. Besides, since the viewer location is dispersed and dynamic, providing high-quality VR services is also significantly challenging. Yang *et al.* [17] explored a joint multicast and unicast solution in heterogeneous cloud radio access networks (H-CRAN). The authors formulated the rate-allocation problem as a mixed-integer nonlinear optimization and proposed two approximate solutions to solve it. By applying a greedy and approximate solution, the proposed approach achieves a near-optimal performance with low time complexity. Yang *et al.* [18] focused on the bottleneck of VR performance and have tried to reduce the resource overhead of VR games on client devices. The authors have also improved an open-source testbed, called Air Light Virtual Reality (ALVR), for cloud-based VR gaming and have measured the performance of real players under different network conditions. Since the cloud is geographically remote from the viewers, live VR services are delay-sensitive and vulnerable to high delivering latency. Edge computing, located closer to the user, is an approach proposed by many researchers for live VR video processing [4, 5, 7, 8, 10, 19–28]. For instance, Guo *et al.* [4] proposed an adaptive VR framework for efficient real-time VR video rendering by offloading the processing tasks to mobile edge computing servers. This solution utilized collaboratively MEC servers and mobile devices to render the foreground and background of VR video for the viewers. The authors of [7] presented a novel online Nash reinforcement learning-based solution to achieve a good trade-off between bandwidth-related performance and resource utilization in 5G HetNet environments with D2D communications, resulting in a 50% performance improvement under a moderate resource cost. L. Liu *et al.* in [25] considered the convergence of communication and computing and proposed a new fog computing-enabled mobile network framework that supports various wireless multimedia services. The authors of [26] introduced an edge-based solution to enhance multimedia services by integrating data processing and distribution into the network functionality. In addition, Argyriou *et al.* [10] provided a novel MEC-assisted transcoding framework for improving cost efficiency by smart allocation of viewpoint rendering to mobile edge computing services. The authors formulated the resource allocation problem as a multi-objective combinatorial optimization problem with delay constraints to provide ultra-high resolution and ultra-low latency VR services. They also proposed a transmission optimization algorithm to maximize user QoE and minimize system overhead. Additionally, the authors of [27, 28] proposed using edge caching and field-programmable gate array (FPGA) as edge computing devices to facilitate efficient video processing and achieve rapid response, while also enabling energy consumption reduction.

Since VR video processing requires personalized VR video preparation for each client, data delivery is a critical issue for supporting high-quality VR live services. Multicast is a popular transmission technique for live video streaming in 5G-HetNets [9, 11, 39], but it is challenging to be used to deliver personalized content. Next, some existing multicast



(a) The overlap of different videos under different frames

(b) Delay vs time in different locations and devices

(c) The number of concurrent users over time

Fig. 2. Analysis of viewer behavior in VR service

VR delivery solutions are discussed.

2.2 VR Multicast over 5G-HetNets

As wireless networks employ broadcast communications at their core, multicast technology for VR streaming in 5G-HetNets is increasingly being studied [9, 11, 29–31, 39–41]. For example, Guo *et al.* [39] investigated tiled VR video multicast from one base station to multiple clients. To minimize the transmission cost while maximizing QoE, the authors formulated a joint optimization as a non-convex problem. Moreover, they designed a greedy-based algorithm that can obtain a near-optimal solution. Guo *et al.* [41] further transform the joint optimization problem into an equivalent convex problem and design two optimal closed-form solutions to consider viewing behavior and dynamic network conditions. These solutions offer a new avenue for problem optimization by exploring the original problem structural properties. Long *et al.* [9] considered multi-quality tiled VR video streaming in 5G-HetNets and introduced two new types of multicast solutions named smoothness-enabled multicast and transcoding-enabled multicast. In their work, the authors also presented a novel mathematical model to determine the effect of multicast on transmission and transcoding cost consumption. To alleviate the traffic load of wireless networks, the authors of [11] proposed a dynamic adaptive streaming solution over HTTP (DASH) based tiled multicast solution with a weighted tile approach and a rate adaptation algorithm. The results presented showed that the solution effectively reduces bandwidth consumption and improves QoE compared to traditional multicast solutions. Bao *et al.* [42] presented a motion-prediction-based multicast for concurrent viewers servicing to optimize the wireless bandwidth utilization and achieve efficient transmission of live 360-degree video content. The authors collected viewing traces of more than 150 users, which revealed a similarity of user viewing motion patterns, and proposed a trace-driven multicast strategy to lower bandwidth consumption.

Multicast is a promising avenue for live VR video services in 5G-HetNets. To prove this point, we have analyzed user viewing overlap for three types of videos from the dataset [43]. As omnidirectional videos have an obvious hotspot for most viewers, multicast is indeed an effective solution for live VR transmissions. However, most of the current live VR solutions [4, 5, 7–11, 14–31, 39–42] ignore the asynchronism of geo-distributed users in content delivery. We collected the geo-distributed *broadcaster-to-viewers* delay and viewer variation with a web crawler[4]. Further, we analyzed 48 different users' viewpoint overlap/locations based on the public dataset [43]. Fig. 2 (a) illustrates the viewpoint overlap for different frames. Fig. 2 (b) gives the stream latency variation for different devices. We note that the delay of geo-distributed viewers across the globe (i.e. Los Angeles, Waterloo, Hong Kong, Singapore, Tokyo) equipped with different devices (tethered and wireless) varies a lot. This phenomenon proved the time-shift phenomenon and revealed that the immersive performance of current live streaming platforms is not good. We also found that the number of users is often very dynamic on live platforms. Fig. 2 (c) shows the concurrent audience variation on a famous professional match from Tencent League of Legends. Nam et al.'s research results show that rebuffering events are much more noticed and annoy viewers more than the start-up latency and bitrate changes in the context of live streaming services [44]. Another study of Rainer et al.[45] noted that buffer-based adaptation logic performed better in terms of average bitrate than the rate-based adaptation results. Hence, we propose a buffer-based multicast solution to capture the time-shift of different users and provide a high-quality immersive viewing experience by balancing and maximizing the buffer level of each user.

Compared with existing solutions, our work has several significant differences and advantages. First, we build a novel buffer evolution model by quantifying the impacts of data transmission and user playback behavior on the change of user buffer size in discrete time. Secondly, we formulate the multicast data schedule problem as a buffer-nadir maximization problem which considers the balance of buffer sizes between different users to achieve the fairness of video buffering and further solve the time shift problem. Further, we consider the joint optimization of delivery and transcoding and formulate it as a stochastic problem to alleviate resource consumption. Finally, we propose two approximate optimum algorithms called MATO and CAD, with only a constant additional factor, which achieve throughput improvement, delay reduction, and resource-saving.

3 MEC-DC Framework and System Model

This section shows the **M**ulti-user cost-**E**fficient **C**rowd-assisted **D**elivery and **C**omputing (MEC-DC) framework and system model in 5G HetNets. Note we use lowercase

[4] Python-built Crawler for Twitch: https://github.com/uglyghost/simple_crawler_for_twitch

TABLE 1
Mathematical Notations

Symbol	Description
$\mathcal{B}$	A live VR content provider
$\mathcal{V}$	A set of viewer clusters
$s_0, \mathbb{S}_s$	A macro-based station (BS) and a set of small BSs
$b_0, \mathbb{B}_s$	Backhaul capacities of macro BS and small BSs
$\mathbb{S}$	A set of all base stations including MBS and SBSs
$\mathbb{V}, v_i$	Live VR content library and specific content i
$M, \mathbb{F}$	The number of tiles and resolution set of each tile
f_h, λ_h	A specific resolution and its transcoding cost
$c_0, \mathcal{C}_s$	Computing capacities of macro BS and small BSs
$\mathcal{T}$	A set of the time-slotted system
d_n	Downlink bandwidth of the n-th BS
γ^n	Index of the multicast decision for n-th BS
Ω	A joint set of the indexes for multicast decision
$\boldsymbol{\gamma}_{n^\dagger}$	A multicast policy of the $n^\dagger$-th BS
$\boldsymbol{\pi}_{n^\dagger}$	A transcoding policy of the $n^\dagger$-th BS
$\mathbb{H}_{n^\dagger}[t]$	A set of wireless channel state of the $n^\dagger$-th BS at t
$u = (v_i, f_j, s_n)$	multicast cluster for the j-th resolution of the i-th content in the n-th BS.
$h_u[t]$	A transmission identifier for multicast cluster u at t
$\chi_u[t]$	Playback buffer of the cluster u at t
$\bar{\chi}_u$	The average value of buffer-nadir for all clusters
τ	Upper bound of the buffer level
$\mathbf{\Psi}$	Feasible region of the policy γ
ϕ	Auxiliary value
P	Services probability from different nodes (MBS, SBS)
$\mathcal{J}_n'(\boldsymbol{\pi})$	Total transmission cost of all BSs
λ, α, β	Weights of resource cost for different type of nodes
$\boldsymbol{q}_u[t]$	Virtual queue-length of multicast cluster u at t
V	The weight value of virtual queue update

italic symbols as scalars. Lowercase *italics* **bold** type and uppercase *italics* typo to indicate vectors and sets, respectively. Uppercase, *italics* **bold** fonts represent matrices. The tarefnotations lists all mathematical notations used in this paper.

3.1 MEC-DC Framework

Fig. 3 illustrates the MEC-DC framework for live VR content delivery in a 5G HetNet environment. The framework considers multiple types of nodes, including a live VR content provider $\mathcal{B}$, a set of viewer clusters $\mathcal{V}$, a macro-cell based station (MBS) s_0 and N small-cell based stations (SBS), denoted by $\mathbb{S}_s = \{s_1, ..., s_N\}$. SBSs are uniformly distributed over MBS's range and connect to MBS with optical fibers, whose bandwidth capacities are denoted by $\mathbb{B}_s = \{b_1, ..., b_N\}$. First, we give some general assumptions. We assume that MBS is connected to the live VR platform via a b_0 bandwidth backhaul link and can communicate to all viewers over HetNet, while SBSs can only be associated to viewers in their coverage area [46]. For simplicity, we assume that the coverage between SBSs, and the operating frequencies of MBS and SBSs are non-overlapping [46]. In other words, the users can access at most one SBS at a time, but can be served concurrently by both MBS and SBSs.

The proposed MEC-DC system considers three stages, namely cooperative processing, cooperative delivering and buffer evolution. During **cooperative processing**, the VR content providers continuously generate and upload original live VR streams to the delivery system. Afterwards, the system processes the multiple VR live streaming to tile-based content, and delivers them to MBS and SBSs. Additionally, MBS further manages all base stations and assigns processing tasks, such as video transcoding, to SBSs. BSs $\mathbb{S} = s_0 \cup \mathbb{S}_s$ will process the assigned tasks cooperatively and will deliver the transcoded tiles of appropriate resolutions to the users according to their FoVs and configurations. Once the tiles arrive at the users' sides, the client stitches the received tiles into sphere frames and then video segments. During **cooperative delivery**, BSs transmit the tile-based VR content to the users via multicast. Additionally, users can directly communicate with each other using D2D communications. In the MEC-DC framework, MBS, as a controller, is responsible for global data delivery scheduling in units of tiles. MBS determines its transmission strategy first, then for SBS, and third, it enhances the data delivery by using D2D communications. Finally, in order to address the synchronization of user playback, we consider and model the **buffer level evolution** of each user with time. Buffering occurs when the user retrieves all the tiles for a frame and completes the stitching process. In the following subsection, we provide more details about the model.

We consider a live VR content library consisting of C different live VR pieces of content denoted by $\mathbb{V} = \{v_1, v_2, ..., v_C\}$. We assume that the popularity of content follows Zipf's distribution [47, 48] with popularity exponent ξ and that the MEC-DC system processes 360 degree videos with equirectangular projection. Thus, each VR video can be further divided into multiple rectangular tiles, as shown in the cooperatively processing part of Fig. 3. We define that each piece of content has M number of tiles and each tile has K different resolutions, denoted as $\mathbb{F} = \{f_1, f_2, ..., f_K\}$. We define the highest representation as f_1 and the lowest as f_K, so we have $f_1 > f_2 > ... > f_K$. According to [9, 10], we assume that each tile of the streamed source can be transcoded to multiple versions and define λ_h as the transcoding cost of version f_h, where $h \in \{1, 2, ..., K\}$. We express the MBS and SBSs computing resources which can be used for video transcoding as c_0 and $\mathcal{C}_s = \{c_1, c_2, ..., c_N\}$, respectively. Due to the resource limitation of edge nodes, BSs can only support a certain amount of transcoding workload. When the edge nodes are fully loaded, they need to access the target version of tiles from the upper server (*i.e.*, cloud servers of the live VR system) via the backhaul link.

3.2 Buffer Evolution Model

Before presenting the buffer evolution model, we make some preliminary assumptions. We consider a discrete-time system with slots $\mathcal{T} = \{0, 1, ..., T\}$. At each time slot, BSs will jointly decide the multicast policy and adjust the offloading strategy. Without loss of generality, we assume that the downlink bandwidth of the n-th BS is d_n and that BS can process the assigned tasks within their computing capabilities c_n. We assume that $b_n > d_n$, $n \in \{0, 1, 2, ..., N\}$, which means backhaul link is not the bottleneck for the live VR delivery system.

In live VR services, the different viewers' FoVs are also diverse, often with overlapping areas and distinct viewports. The buffer evolution presented in Fig. 3 shows an example of three viewers with different FoVs. As it can be seen, the overlapping FoV areas of viewers include the tiles $\{v_4, v_5, v_{10}, v_{11}, v_{12}\}$, and multicast clusters can be formed according to these tiles. For brevity, we define $\Omega = \{(v_i, f_j, s_n) |\ v_i \in \mathbb{V},\ f_j \in \mathbb{F},\ s_n \in \mathbb{S}\}$ as the joint set of indexes for different multicast clusters and use $u = (v_i, f_j, s_n)$

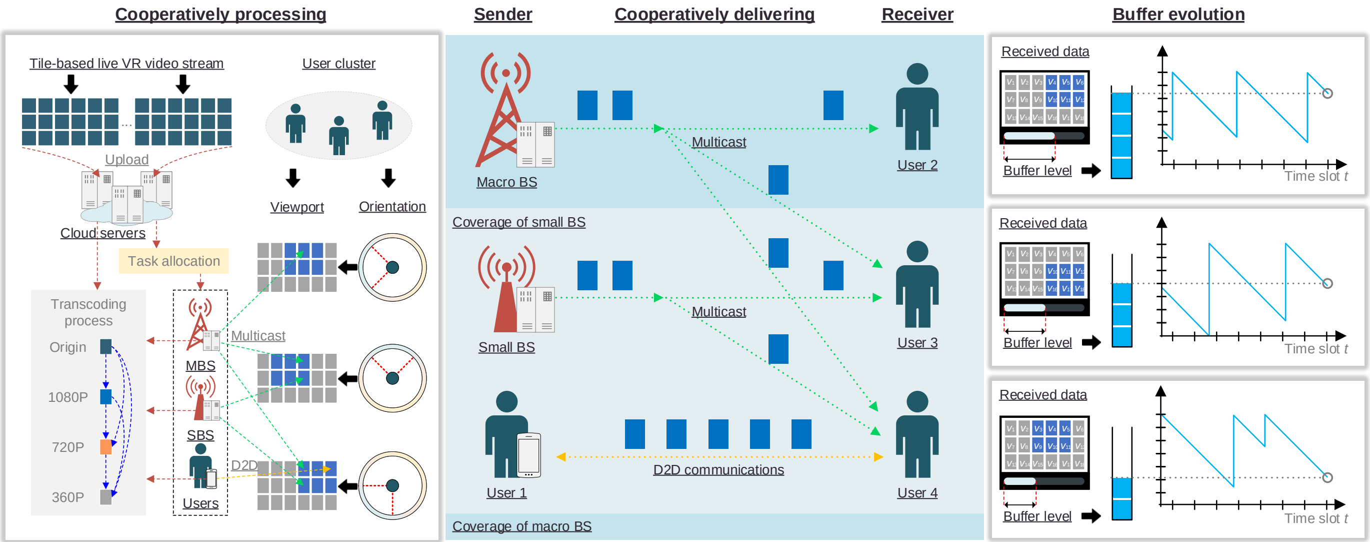


Fig. 3. Illustration of the MEC-DC framework for edge-assisted live VR content delivery in 5G HetNets

to represent the cluster for the j-th resolution of the i-th video tile in the n-th BS. We use a bool variable γ_{ij}^{n} to indicate whether the n-th BS will multicast the video tiles v_i of resolution f_j. Hence, we can express the $n^{\dagger}$-th BS multicast policy $\boldsymbol{\gamma}_{n^{\dagger}}$ as an index vector:

$$\boldsymbol{\gamma}_{n^{\dagger}} = (\gamma_u \in \{0,1\} : u \in \Omega_{n^{\dagger}})$$

where $\Omega_{n^{\dagger}}$ represents the set of all the multicast clusters in the BS $n^{\dagger}$. Besides, we define $\boldsymbol{\pi}_{n^{\dagger}}$ as the transcoding policy of the $n^{\dagger}$-th BS and use the binary variable π_u, $u \in \Omega_{n^{\dagger}}$ to indicate whether the transcoding tasks for multicast cluster u are deployed in the $n^{\dagger}$-th BS. Therefore, we can express the transcoding policy of the $n^{\dagger}$-th BS as the following vector:

$$\boldsymbol{\pi}_{n^{\dagger}} = (\pi_u \in \{0,1\} : u \in \Omega_{n^{\dagger}})$$

Considering tiled live VR services, we use $\mathbb{H}_{n^{\dagger}}[t] = \{h_{u_1}[t], h_{u_2}[t], \ldots\}$ to denote the identifies vector of the multicast clusters within the $n^{\dagger}$-th BS coverage at time t, where $h_u[t]$ identifies whether the tile being sent to the multicast cluster u at time t is the last tile of the latest video segment provided by BS during one time-slot transmission. Thus, we assume that the identifier $h_u[t]$ has two states, YES and NO. When $h_u[t]$ is in YES state at t, $h_u[t] = 1$, otherwise $h_u[t] = 0$. Thus, one buffering process for the multicast cluster u occurs at time t, if and only if $\gamma_u[t]h_u[t] = 1$.

In addition, we define $\chi_u[t]$ as the remaining playback buffer of the multicast cluster u at time t. The evolution of $\chi_u[t]$ is shown in Fig. 4. Buffer $\chi_u[t]$ increases to τ upon a successful transmission, and decreases by 1 in every slot in which there is not any successful activation. Note that the platforms need to wait for the next segment to be generated by broadcasters. The average sending rate of the video is always less than or equal to the average generation rate of VR content in long-term perspective. However, the generation rate of live streaming depends on elapsed time, which means live streaming services have maximum buffer charging. In other words, the receiver can only receive the most recent video content produced by the content provider. Thus, we assume that one successful transmission can replenish the user buffer to τ without video segment loss. The value of $\chi_u[t]$ is updated as follows:

$$\chi_u[t+1] = \begin{cases} \chi_u[t] - 1 & \gamma_u[t]h_u[t] = 0 \quad \text{(1a)} \\ \tau - 1 & \gamma_u[t]h_u[t] = 1 \quad \text{(1b)} \end{cases}$$

where $\chi_u[t]$ represents the buffer for all user of the multicast cluster u and $\gamma_u[t]h_u[t] = 1$ means the cluster receives the VR content successfully. Note that when the user suffers the segment loss during one slot transmission, the user's buffer size increases to an amount less than τ. It indicates that the network condition of the user cannot support the current resolution of tiled VR video, and the user will automatically switch to a lower bitrate multicast cluster. The buffer evolution equation can be rewritten as follows:

$$\chi_u[t+1] = \chi_u[t] - 1 + (\tau - \chi_u[t])\gamma_u[t]h_u[t] \tag{2}$$

When the multicast is successful, the corresponding playback buffer will be charged to τ. We name **buffer-nadir** as the value of the extreme point before each charge. The positions of the buffer-nadir points are shown with red dots in Fig. 4. In this example, the $\chi_u[t]$ can be a negative value to indicate that the playback buffer is empty. When playback buffer is empty, the video player can freeze the video playback, which is called the video playback freeze [49]. Therefore, we can define the average value of buffer-nadir for multicast cluster u as:

$$\bar{\chi}_u = \liminf_{T\to\infty} \frac{\mathbb{E}\left[\sum_{t=0}^{T-1} \chi_u[t]\gamma_u[t]h_u[t]\right]}{\mathbb{E}\left[\sum_{t=0}^{T-1} \gamma_u[t]h_u[t]\right]}, \ u \in \Omega \tag{3}$$

where $\mathbb{E}(\cdot)$ is the expectation. We can define the average buffer-nadir of the total system as $\bar{\chi} = \sum_{\Omega} \bar{\chi}_u$. The next section will introduce the problem formulation. Note that we set the objective to buffer-nadir maximization rather than latency minimization, as performed by other researchers, for instance in[33, 50]. This is as the overall goal is maximization of viewer QoE and not optimization of delivery QoS levels for VR live streaming services. Research [44] has shown that video playback freezes caused by empty buffers are noticed more by viewers and annoys them more than slight latency increases. Authors of [44] collected over 400,000

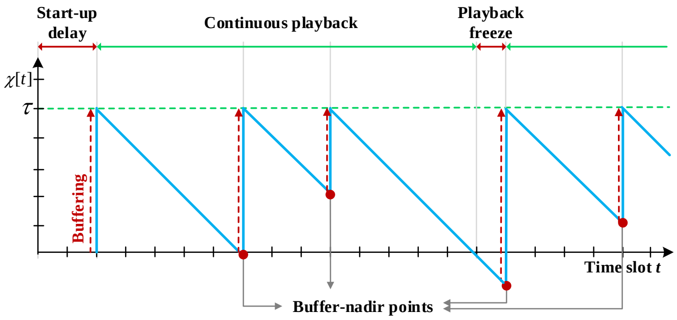


Fig. 4. The evolution of buffer size $\chi[t]$

YouTube viewing records, and their data analysis shows that rebuffering is viewer QoE most important impact factor. Another investigation [45] also demonstrated that a buffer-based adaptation solution for remote distribution of video services performed better than a conventional approach and is more flexible [33, 50].

4 Problem formulation

This section presents the formulation of the cooperative multicast scheduling as a buffer-nadir maximization problem. We further discuss the resource consumption of task allocation for heterogeneous edge networks and formulate the multicast-aware transcoding offloading problem.

4.1 Buffer-Nadir Maximization Multicast

Before introducing the problem formulation, we first confine our discussion to a reasonable scope. We consider that our policy is consistent with the following assumption:

$$\boldsymbol{\Psi} = \{\boldsymbol{\gamma} \mid \exists\theta\ s.t.\ \tau > \bar{\chi}_u(\boldsymbol{\gamma}) \geq \theta,\ \forall t > 0,\ u \in \Omega\} \tag{4}$$

This equation implies that we only consider the policy $\boldsymbol{\gamma}$ that makes the average buffer-nadir of each channel not less than a positive value θ, where $0 < \theta < \tau$. This restriction limits our policy to the set which can provide effective live VR services for viewers. Based on this, we give an essential lemma that is always true for policy $\boldsymbol{\gamma} \in \boldsymbol{\Psi}$ and present the proof in Appendix A.

Lemma 1. *For all the policies $\boldsymbol{\gamma}$ that belong to $\boldsymbol{\Psi}$, the following equation is satisfied:*

$$\bar{\chi}_u = \tau - \frac{1}{\limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\gamma_u[t]h_u[t]\right]} \tag{5}$$

where $u \in \Omega$.

In addition, we assume that MBS has global information about all viewers' requests and divides users according to their requested tiles. The users with requests for the same tile will be grouped into one multicast cluster. Further, we consider the multicast cluster as the basic unit to discuss the problem formulation and we focus on the buffer-nadir maximization problem for the policy space $\boldsymbol{\Psi}$. During live VR delivery, increasing the buffer-nadir avoids the playback freeze and improves viewer QoE. Thus, we define the optimal buffer-nadir as $\bar{\chi}^* = \max_{\boldsymbol{\gamma}\in\boldsymbol{\Psi}} \bar{\chi}(\boldsymbol{\gamma})$ and our goal is to maximize $\bar{\chi}(\boldsymbol{\gamma})$ without violating the resource constraints. Therefore, the average buffer-nadir maximization problem can be formulated as follows:

$$\text{Max} \sum_{u\in\Omega} \bar{\chi}_u \tag{6a}$$

$$s.t. \ \liminf_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\gamma_u[t]h_u[t]\right] \geq \phi_u,\ u \in \Omega \tag{6b}$$

where ϕ_u is the auxiliary value of multicast cluster u. We can derive (6b) and $\frac{1}{\tau-\theta} \leq \phi_u \leq 1$ based on (3), (4), (5). We note that the condition implied by eq. (1) indicates that one successful transmission can only replenish the user buffer size to τ at most. Since our objective function is to maximize the buffer-nadir of all users equally, the benefit of filling the buffer for different users is related to their instant buffer size. In other words, servicing a user whose buffer is about to run out is better than a user whose buffer is almost full. Therefore, our goal is to balance buffer size between different users, avoid having viewers' buffers empty, and address the time-shift problem.

4.2 Multicast-Aware Transcoding Offloading Problem

Since MBS can obtain global information and serve all viewers, it acts generally as the central controller in a cell [51]. In the system, MBS makes its own decision first, and then schedules the resources of SBSs to provide services to viewers. We define the service probability $P = \{p_0[t], p_{n^\dagger}(\boldsymbol{\pi}_{n^\dagger}[t]), p_r(\boldsymbol{\pi}_n[t])\}$ for the clusters of n-th BS with different sources (MBS, n-th SBS and remote servers) that satisfy $p_0[t] + p_n(\boldsymbol{\pi}_n[t]) + p_r(\boldsymbol{\pi}_n[t]) = 1$. Because we assumed that the backhaul bandwidth was always abundant ($b_n > d$) in the previous section, the multicast policy of the n-th BS $\boldsymbol{\gamma}_n[t]$ is independent, with offloading strategy $\boldsymbol{\pi}_n[t]$. Thus, the total transmission cost of all BSs can be written as follows:

$$\mathcal{J}'_n(\boldsymbol{\pi}) = \sum_{t=0}^{T}\sum_{u\in\Omega_n}\gamma_u[t]\Big(\alpha_u p'_n(\pi_u[t]) + \beta_u p_r(\pi_u[t])\Big) \tag{7}$$

where α_u and β_u are the weight factors of bandwidth cost from base station and remote server, respectively. Intuitively, access content from remote servers has the highest resource consumption β_u which is indicated $\alpha_u < \beta_u$.

The first term $\alpha_u p'_n(\pi_u[t])$ on the right-hand side of Eq. (7) represents the local multicast consumption of the n-th BS under policy $\boldsymbol{\pi}_n[t]$ and $p'_n(\pi_u[t]) = p_n(\pi_u[t]) + p_0[t]$. In other words, the first term consists of two parts: cost of the transmission from MBS and from SBS, respectively. The second term $\beta_u p_r(\pi_u[t])$ includes the transmission cost of multicast and acquisition cost of VR content from the remote server. We consider the local multicast as a two-step process. In each slot, MBS multicasts the video segments with priority, then SBSs deliver the content to their local clusters. Hence, the expressions of services' probability for SBS n are:

$$p_n(\pi_u[t]) = \begin{cases} 0, & \pi_u[t] = 0 \quad (8a) \\ 1 - p_0[t], & \pi_u[t] = 1 \quad (8b) \end{cases}$$

Similarly, we have the probability p_r of the remote server:

$$p_r(\pi_u[t]) = 1 - p_0[t] - p_n(\pi_u[t]), \tag{9}$$

This equation means that when the n-th SBS has the VR content demanded by multicast cluster u, i.e., $\pi_u[t] = 1$, the probability $p_n(\pi_u[t])$ of the cluster user u of being served by the n-th SBS is equal to $1 - p_0[t]$. Otherwise, $p_n(\pi_u[t])$ is equal to 0 and the remote server will serve the cluster u with probability $1 - p_0[t] - p_n(\pi_u[t])$.

The Multicast-aware transcoding offloading problem determines the offloading policy for all BSs that minimize the expected resource cost in the whole period. Thus, the problem can be formalized as follows:

$$\text{Min } \mathcal{J}(\boldsymbol{\pi}) = \sum_{n=0}^{N} \left(\sum_{t=0}^{T} \sum_{u \in \Omega_n} \lambda_u \pi_u[t] + \mathcal{J}_n'(\boldsymbol{\pi}_n[t]) \right) \quad (10a)$$

$$s.t. \sum_{u \in \Omega_n} \lambda_u \pi_u[t] \le c_n, \ \forall s_n \in \mathbb{S} \quad (10b)$$

$$\sum_{u \in \Omega_n} \alpha_u \gamma_u[t] \le b_n, \ \forall s_n \in \mathbb{S} \quad (10c)$$

where λ_{ij}^n is the weight factor and the first term of the objective function $\sum_{t=0}^{T} \sum_{u \in \Omega_n} \lambda_u \pi_u[t]$ is the transcoding cost of the n-th BS. The constraints from eq. (10b) ensure that the transcoding workload of the n-th BS will not exceed the capacity. Eq. (10c) indicates that the transmission bandwidth cannot exceed the link capacity.

Lemma 2. *We define transcoding priority as* $\psi_u[t] = \frac{(\beta_u - \alpha_u)\gamma_u[t]}{\lambda_u}$, $u \in \Omega_n$ *and the n-th BS only needs to transcode the content that satisfies the following inequation:*

$$\psi_u[t] \ge \frac{1}{1 - p_0} \quad (11)$$

BSs prefer to transcode the content with higher $\psi_u[t]$, *which often results in a lower cost-effectiveness ratio.*

The proof of **Lemma 2** is shown in Appendix B.

5 Joint Optimization Algorithms

This section first discusses the joint optimization problem of multicast and transcoding offloading. Afterwards, in order to solve the problem, we design two algorithms, which iteratively optimize the multicast scheduling and transcoding allocation in a decentralized fashion.

5.1 Optimization of Multicast and Task Offloading

This sub-section presents the optimal solution of multicast and task offloading. Inspired by [52], we first provide the optimal buffer-nadir solution, which is called the h-only policy. In the 5G HetNets, the wireless channel state for different multicast cluster is often stochastic and equivalent. Since the arrival of viewer request is i.i.d. with respect to time t, the transmission process $\gamma[t]h[t]$ is also i.i.d. across time t under the policy $\boldsymbol{\gamma} \in \boldsymbol{\Psi}$. We denote ε as the expectation of the stochastic process $\gamma[t]h[t]$. Thus, according to **Lemma 1**, we rewrite the buffer-nadir maximization problem from eq. (6a) as follows:

$$\text{Min} \sum_{u \in \Omega} \frac{1}{\varepsilon_u} \quad (12a)$$

$$s.t. \limsup_{T \to \infty} \mathbb{E}\left[\frac{1}{T} \sum_{t=0}^{T-1} \gamma_u[t] h_u[t] \right] \ge \varepsilon_u, \ u \in \Omega \quad (12b)$$

where $\varepsilon_u = \mathbb{E}[\gamma_u[t]h_u[t]]$. Because the optimality of the h-only policy is only influenced by ε_u and we can achieve the δ-optimal solution (12) by following the steps of theorem 4.5 in [52], where δ is an arbitrarily small value, but greater than zero. We denote the h-only policy $\boldsymbol{\gamma}^*$ as the optimal solution for the multicast problem $\text{Min} \sum_{u \in \Omega} \varepsilon_u^{*-1}$.

Next, we consider the optimal transcoding policy $\boldsymbol{\pi}^*$. If we fix the optimal multicast policy $\boldsymbol{\gamma}_n^*$ of the n-th BS, the corresponding transcoding priority $\boldsymbol{\psi}_n(t)$ is also determined. We denote $\boldsymbol{\psi}_n^*$ as the transcoding priority of the n-th BS under policy $\boldsymbol{\gamma}_n^*$. Based on **Lemma 2**, we give the optimal transcoding strategy $\boldsymbol{\pi}_n^*$ as follows. Each BS will search the combination of transcoding tasks with maximum $\sum_{u \in \Omega_n} \lambda_u \psi_u[t]$ under the resource constraints from eq. (10b). This combination ensures minimal resource cost of each BS. Since the consumption of each BS is independent, the above mechanism is the optimal offloading decision $\boldsymbol{\pi}_n^*$. However, the problem is a typical 0-1 knapsack problem, which is NP-Hard. Obtaining the optimal solution to such a problem is often impractical. Therefore, it is necessary to design a lightweight algorithm to solve it.

5.2 Multicast-aware Transcoding Offloading Algorithm

We now design a policy that solves the joint optimization problem of multicast from eq. (6a) and transcoding offloading from eq. (10a). Based on our previous analysis, the buffer-nadir optimization is only determined by the multicast decision and the cost consumption is influenced by both multicast and transcoding offloading. However, excessive playback freezes ($\chi[t] < 0$) are often less acceptable compared to increased cost consumption for live VR services. Thus, we give priority to the effect of buffer-nadir optimization in the design of the algorithm, and further reduce the total cost consumption through task offloading. We introduce the proposed joint optimization algorithm of multicast and task offloading (MATO) in **Algorithm 1**.

Algorithm 1 is deployed at the base station side. We first select the condition parameters. For the iteration phase, according to the differential coverage and problem domain, we divide it into two steps in two parts which are the upper step (MBS) and the lower step (SBSs). Each step includes schedule decisions and offloading decisions. At the beginning of every slot, MBS will make the decisions first. MBS observes the current virtual queue-length of each cluster and selects the multicast cluster u with the maximum queue backlog based on eq. (13). Then, according to the cluster u requirements, MBS multicasts the tiles of the VR video to the users who have the most pressing needs for video buffering. To quantify the degree of urgency for buffering, we introduce a new mathematical concept denoted virtual queue $\boldsymbol{q}[t]$ as in eq. (15). The virtual queue decreases by $\gamma[t]h[t]$ after the viewer downloads a new video segment, and increases by $\sqrt{\frac{V}{q[t]}}$ in each time-slot where V is a weight constant. Note that we set the queue growth as $\sqrt{\frac{V}{q[t]}}$ for two reasons: 1) the queue growth needs to be less than the expected dequeue rate $\mathbb{E}[\gamma[t]h[t]]$. 2) we need to construct a square term to simplify the inequality of Lyapulov's function during the proof of Theorem 1 in Appendix C.

In addition, SBS needs to decide the multicast strategy γ_n according to eq. (14) in the lower step. Different from the

Algorithm 1: MATO Algorithm

```
/*Algorithm processed in bases station side*/
Input:
Choose the condition numbers $\alpha_u$, $\beta_u$ and $\lambda_u$;
while $t \in \mathcal{T}$ do
    Set $c_n^r = c_n$ and $b_n^r = b_n$, $\forall s_n \in s_0 \bigcup \mathbb{S}_s$.
    foreach Base station $s_n \in s_0 \bigcup \mathbb{S}_s$ do
        /*multicast decisions:*/
        if $s_0$ then
            $$\gamma_0[t] = \operatorname*{argmax}_{\gamma \in \boldsymbol{\Psi}} \boldsymbol{q}_u[t] \quad (13)$$
        else if $s_{n^\dagger} \in \mathbb{S}_s$ then
            $$\gamma_{n^\dagger}[t] = \operatorname*{argmax}_{\gamma \in (\boldsymbol{\Psi}/\gamma_0)} \boldsymbol{q}_u[t] \quad (14)$$
        end
        /*task offloading decisions:*/
        Sort the multicast cluster set $\Omega_{n^\dagger}$ in
          descending order of $\psi_u[t]$, get $\Omega_{n^\dagger}^d$
        foreach multicast cluster $u \in \Omega_{n^\dagger}^d$ do
            if $\psi_u[t]$ satisfies Lemma 2 and $c_{n^\dagger}^r > \lambda_u$
             then
                Offload the transcoding task to $s_{n^\dagger}$;
                Set $c_{n^\dagger}^r \leftarrow c_{n^\dagger}^r - \lambda_u$
            end
        end
        Get transcoding policy $\boldsymbol{\pi}_{n^\dagger}[t]$.
    end
end
```

upper layer, the feasible space of SBSs is $\frac{\Psi}{\gamma_0}$, which means that the multicast tile set of MBS and SBS are mutually-exclusive. In the task offloading decision, n-th BS sorts the multicast cluster of set Ω_n in descending order of $\psi_u[t]$. The central MBS will assign the tile transcoding tasks to SBSs in a greedy manner. When a user requests a specific tile, and BS does not transcode it, BS fetches it via backhaul and delivers it to the users. Afterwards, BS removes the allocated transcoding task that does not satisfy **Lemma** 2.

According to the algorithm description, MATO includes two parts: (1) multicast decisions and (2) task offloading decisions. In the first part, the base station selects the multicast policy that maximizes the virtual queue $q_u[t]$. According to the definition from section 3.2, policy $\gamma_n[t]$ needs to decide the multicasting content and the corresponding resolution. Assuming the total number of VR content items and the number of resolutions are C and K respectively, the complexity of the multicast decision is $\mathcal{O}(CK)$. The complexity of the second part of the algorithm is similar to that of multicast decision. Offloading policy is also close related to the amount of transcoding content and its resolution, the complexity is also $\mathcal{O}(CK)$.

5.3 Crowd-assisted Delivery Algorithm

The crowd-assisted delivery and bitrate adaptive process are described in **Algorithm 2**. This is deployed at the user side. For each multicast cluster u, we choose the parameters and set the initial value of the buffer-nadir and the virtual queue-length to 0. Based on the multicast process, the multicast cluster will update the virtual queue-length $q_u[t]$ based on eq. (15), where $\lceil \cdot \rceil^{+1} = \max\{\cdot, 1\}$. Since viewers' devices are often equipped with a FoV predictive model [12], the required tiles have usually different resolutions and are few seconds ahead of the tiles that the viewers are watching. We assume that the users adopt a basic buffer-based adaptive mechanism [45] for live VR video streaming. The user determines the tile resolution f based on the buffer level $\chi[t]$. In the buffer-based strategy, viewer switches to a lower video resolution at time t when $\chi[t]$ is less than the cluster u buffer level $\chi_u[t]$, and requires higher resolution tiles when $\chi[t]$ is greater than $\chi_u[t]$. The inequation $\chi[t] < \chi_u[t]$ means the viewer suffering the segment loss during the cluster-based multicast process. Moreover, the inequation $\chi[t] \geq \chi_u[t]$ indicates that the viewer has sufficient bandwidth resource. The buffer-based tiled dynamic adaptive streaming solution can be formulated as in eq. (16). In addition, the user can also access the content from other nodes in the multicast cluster. Therefore, when the network condition between the user and BS is not good, the content can also be obtained from the cluster via D2D communications.

The theoretical optimality of our algorithms is introduced in **Theorem 1**. Next, its distributed implementation is discussed.

Theorem 1. *The lower bound of the buffer-nadir for our algorithm is*

$$\bar{\chi}(\gamma) \geq \bar{\chi}^* - \frac{(V+1)|\Omega|}{2V\bar{\chi}^{*2} + (V+1)|\Omega|} \tag{17}$$

where $|\Omega|$ is the number of multicast clusters and $\bar{\chi}^* = \frac{1}{\xi^*}$ is the optimal buffer-nadir which is obtained by the ξ-only policy.

The proof of **Theorem 1** is in Appendix C.

Implementation of the Algorithm: In our algorithm, MBS needs to inform all SBSs of its multicast decision $\gamma_0[t]$ in advance, which requires a one-hop information exchange. The multicast decisions of each BS require the virtual queue-length information $q[t]$ of every cluster within its coverage. According to **Lemma 2**, the offloading decisions only need one time-slot of local multicast decision $\gamma[t]$ to calculate $\psi[t]$ of each cluster. Thus, the implementation of our algorithm is straightforward. The time complexity of our algorithm is $\mathcal{O}(CK)$ in BS, where C and K represent the total number of VR contents and resolutions, respectively.

Algorithm 2 is performed at the user device side and there are two steps for each cluster: (1) virtual queue update and (2) crowd-assisted unicast delivery. In the first step, the virtual queue updates once per time slot based on eq. (15), so the complexity is $\mathcal{O}(1)$. During the unicast delivery process, each viewer has to search for which node has the content it demands. Assuming the total number of viewer nodes is U , the maximum number of nodes that need to be searched in the worst case is $U-1$ and therefore, the complexity of the unicast delivery is $\mathcal{O}(U-1)$.

In our solution, the multicast decision in eq. (13) and eq. (14) requires the queue-length information of every cluster within its coverage. The queue update process in each user device also requires the multicast decision according to eq.

Algorithm 2: Crowd-assisted Delivery Algorithm

```
/*Algorithm proceed in user device side*/
Input:
Choose the condition numbers $V$ and $\tau$;
For each multicast cluster $u$. we set the initial virtual
queue-length $\boldsymbol{q}_u[0] = \boldsymbol{1}$ and buffer $\chi_u[0] = 0$, $u \in \Omega$.
while $t \in \mathcal{T}$ do
    foreach multicast cluster $u \in \Omega_{n^\dagger}$ do
        /*virtual queue update $q_u[t]$*/
        $$q_u[t+1] = \left[ q_u[t] + \sqrt{\frac{V}{q_u[t]}} - \gamma_u[t] h_u[t] \right]^{+1} \quad (15)$$
        foreach viewer in multicast cluster $u$ do
            /*Buffer-based adaptive streaming:*/
            Change the resolution $f[t]$ of tiles based
              on the viewer's buffer size $\chi[t]$:
              $$f[t+1] = \begin{cases} f_{\lfloor i-1 \rfloor^1}, & \chi[t] < \chi_u[t] \quad (16a) \\ f_{\lceil i+1 \rceil^K}, & \chi[t] \geq \chi_u[t] \quad (16b) \end{cases}$$
              The viewer switches its multicast cluster
              by updating $f[t+1]$ to MBS.
            /*Crowd-assisted unicast delivery:*/
            if Neighbor nodes have demand tiles or
              higher-resolution version then
                Request demand content from
                  neighbor nodes by D2D
                  communication;
            else
                Request the content from BSs;
            end
        end
    end
end
```

TABLE 2
Bandwidth Requirement and Transcoding Cost

resolution	1080p 60fps	1080p	720p 60fps	720p	480p	360p
bandwidth (Mbps)	5.86	4.45	2.75	1.93	1.10	0.52
vCPU usage	454%	333%	210%	142%	81.6%	50.5%

(15). Therefore, algorithm 1 and algorithm 2 are inter-related and influence each other.

6 Performance Evaluation

This section first introduces the experimental scenario and parameter settings. Then, numerical results when employing our algorithm are analysed. Numerical results prove the validity of **Theorem 1**. Finally, we evaluate the service performance and cost consumption of our algorithm by comparing it with three state-of-the-art solutions [7], [31] and [38].

6.1 Experimental Setup and Datasets

To evaluate the performance of our proposed method, first we carried out a series of numerical simulations. In these simulations, we consider a two-layer HetNet structure in a 1000*2000m^2 scenario with 1000 viewers, which includes a remote cloud server (CS), a MBS, and 10 SBSs with uniform distribution. We assume that the movement behavior of mobile nodes follows the Random Way Point (RWP) model, which is a general mobility model used in mobile network research [47, 48] The square is divided into 10 equal regions, and in each square region, there is a SBS at its center. For each region, we deploy one SBS, where the SBS in that region is connected to MBS via a backhaul link with 1.0 Gbps. We assume MBS can access all viewers in the scenario and SBSs can serve the users within its square coverage. In addition, MBS is connected to the remote cloud server via a backhaul link. The total number of VR library entries is set to 40. The popularity of the content follows a Zipf-distribution. We consider that the resolution of the VR video source is 8K (7680*4320) and each tile is a full HD video (1920*1080), thus in our simulation each VR video is stitched from 4x4 full HD tiles. According to [53], for tiled VR video, we consider each tile has six resolutions and set up the transcoding expense λ and transmission consumption α of different resolutions. The Table 2 results are measured with an Amazon Web Services instance and a Twitch's official tool. The playback duration of a segment is set to $\tau = 15$ and simulation time is $T = 10^4$ time-slot. The rest of parameters are set as follows: $C = 40$, $d = 1.0$ Gbps, $\beta_u = 3\alpha_u$, $c_0 = 500$ and $c_i = 200$, $i \in [1, N]$. Second, in order to evaluate the service performance of our solution, especially the synchronization, we conducted a system-level evaluation in a prototype based on two real-world user viewpoint datasets [43, 54]. Furthermore, we illustrate some essential details of these two datasets and show how they contribute to our experiments as follows.

- **360° Video Viewing Dataset** [43]: The authors provide a dataset which contains both content data and sensor data. The content data includes saliency maps and motion maps of the VR video. The sensor data includes head positions and orientations trace of 50 viewers which is collected by HMD sensors.
- **Head Tracking Dataset for Spherical VR** [54]: The authors present the head trajectories of 48 users as they watched 18 different videos from 5 categories. The dataset records users' head orientations, users' head movement in each session, and impressive targets of each user in the VR video. Based on the dataset, the authors further present the users' actual viewport which reveals similar viewing behaviours among different users watching the same VR content.

In our prototype-based experiment, we used the head tracking data as user requests to drive our experiment. The data reflects the user FoV and we let the prototype system transmit a high-resolution FoV and a low-resolution background for each individual.

We set the video to start playing when the buffer size reaches τ. First, we evaluate two performance metrics under different V values:. (1) average virtual queue-length $q[t]$ (AVQ). (2) the average buffer-nadir (ABN) evolution. Then,

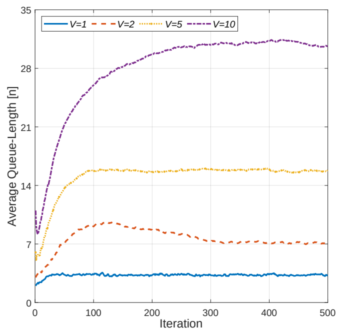


Fig. 5. Average virtual queue-length Evolution.

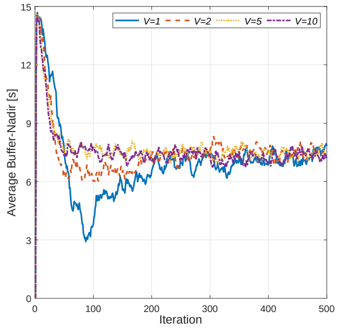


Fig. 6. Average buffer-nadir Evolution.

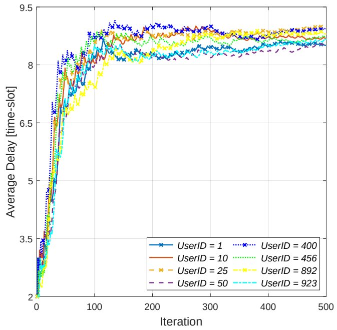


Fig. 7. Average delay of different users.

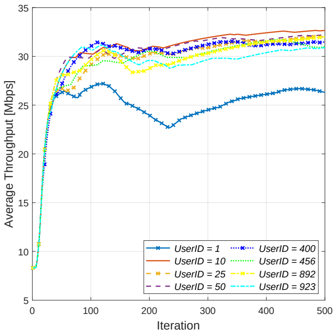


Fig. 8. Average throughput of different users.

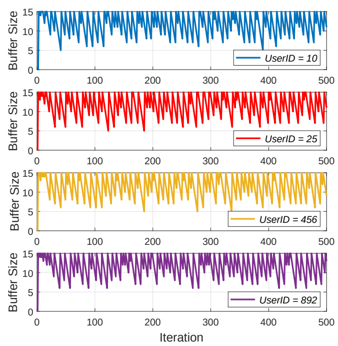


Fig. 9. Buffer size of different users.

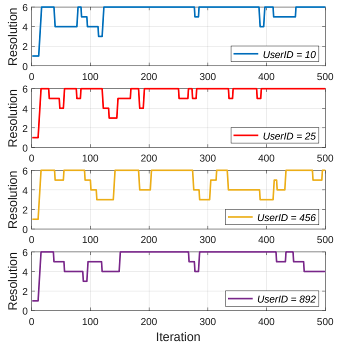


Fig. 10. FoV Resolution evolution.

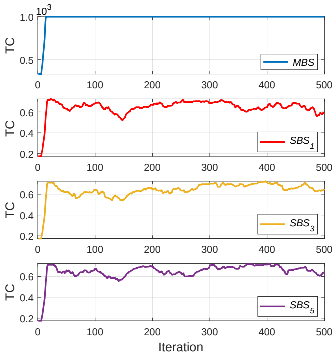


Fig. 11. Transmission cost of different BSs.

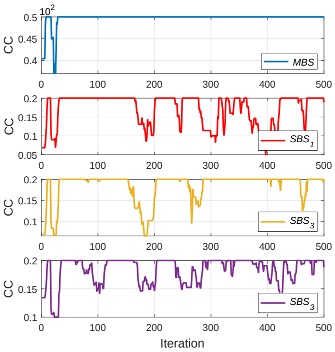


Fig. 12. Computing consumption of different BSs.

we test the QoS performance in terms of the following three metrics. (1) Average Throughput (AT): the average bitrate of successful message delivery over a communication channel. (2) Average Delay (AD): the average latency for data to travel across the network from one communication endpoint to another. (3) Resources Consumption: the resource cost (i.e., bandwidth and CPU usage) for transmission and video transcoding. Additionally, QoE was assessed in terms of the following three metrics: (1) Playback Freeze Ratio (PFR): the ratio of playback freeze time to total video playback time. (2) Start-up Latency (SL): the delay between viewers' access to the VR video and the video actual display. (3) Bitrate Change Ratio (BCR): the ratio of bitrate switching occurred to the total time slot. In addition, we also compute the overall QoE performance, which is expressed according to [55], as follows:

$$QoE = \mathcal{U}(AT) - (w_1 PFR + w_2 SL + w_3 BCR) \tag{18}$$

where $\mathcal{U}(\cdot)$ is the rate-related utility function used in [47] and [48] and we set $w_1 t_P = 6w_2 = 3w_3 t_B$ based on [44], where t_P and n_B are elapsed duration and change size respectively.

6.2 Methodology and Numerical Result

We conduct some numerical studies to verify the theoretical results. To drive our simulation, we set the request generation according to a Poisson distribution and let the viewer requests obey a Zipf distribution. We evaluate AVQ and ABN performance under different V. Fig. 5 presents the Average Virtual Queue-length evolution for all multicast clusters $u \in \Omega$ when V = 1, 2, 5, and 10. We can see that the AVQ first experiences a period of sharp increase at different rate and then reaches different stable values. As V increases, the AVQ in the stable stage grows and the variation becomes larger. The steady-state AVQ is about 3.5, 7, 15.5, and 30.1 when V is equal to 1, 2, 5, and 10, respectively. Fig. 6 shows that the ABN evolution first experiences a descent, and then stabilizes. The stable value of ABN all converges to 8 with different convergence rate when V = 1, 2, 5, and 10, respectively. According to **Theorem 1**, the corresponding theoretical values of Average Buffer-nadir are all converge to $\bar{\chi}^* - 1$ since $|\Omega|$ is usually much larger than χ^*. The experimental results are consistent with the Theorem 1. Based on ξ-only theory, the optimal $\bar{\chi}^*$ is about 7.5 and the numerical results demonstrate the correctness of the theoretical results of our algorithms.

Fig. 7 illustrates the download delay variation of eight concurrent users (UserID=1, 10, 25, 50, 400, 456, 892, 923) requesting the same live VR content. We can see that our solution has good synchronization performance since the user delay has mainly remained under control between the 8.5 and 9 time-slot. Synchronization is especially important for immersive and interactive experiences in multi-user scenarios. Further, we provide the average throughput (AT) variation of different users in Fig. 8. As we can see, the curves first experience a rapid increase and then reach a plateau. In addition, the average throughput of all users (except UserID = 1) converges to about 32Mbps. Combining Fig. 7 and 8, the download delays of the users with different average throughput are almost equal, indicating that our solution provided proximal latency performance for users with different throughput.

Fig. 9-12 reveals more details of our solution numerical results including buffer size variation, FoV resolution evolution, and resources cost (transmission cost (TC) and computing consumption (CC)) of different base stations. From Fig. 9 and 10, we can observe the buffer size variation and FoV

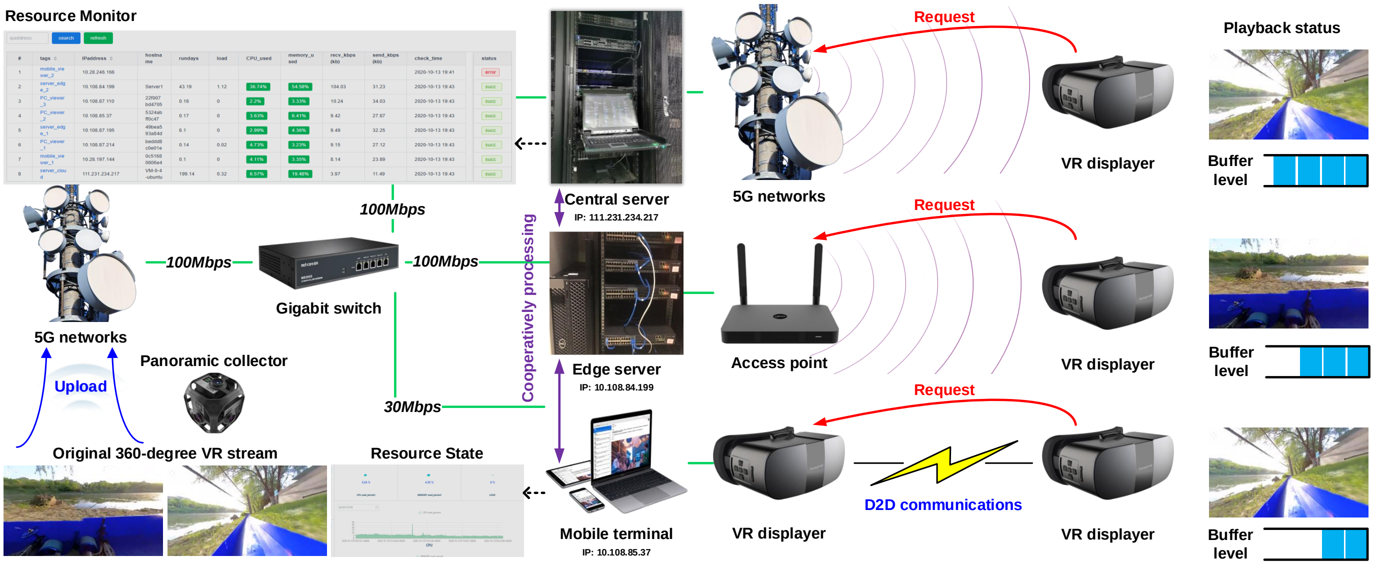


Fig. 13. The topology diagram of the prototype system.

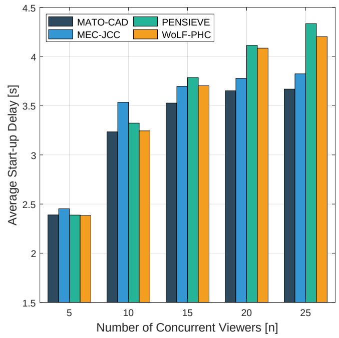


Fig. 14. Start-up latency for increasing number of concurrent viewers.

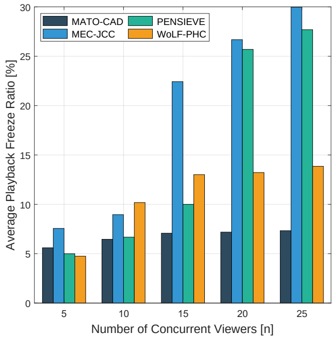


Fig. 15. Playback freeze rate for increasing number of concurrent viewers.

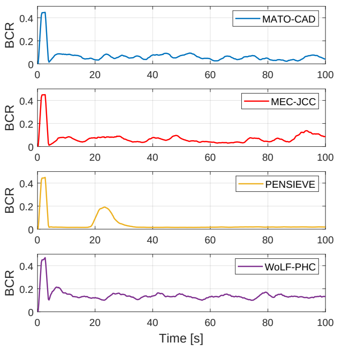


Fig. 16. Bitrate change ratio for the four solutions.

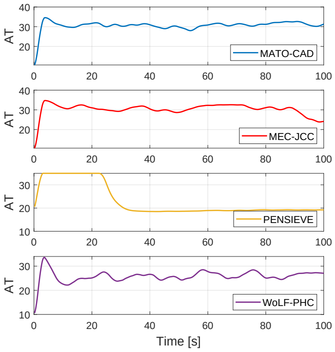


Fig. 17. Average throughput for the four solutions.

resolution evolution of four users (UserID=10, 25, 456, and 892). The buffer evolution of different users is regular, and the buffer-nadir values keep above 5. And even if the users have different representation of FoV range, our solution still maintains stable buffer evolution. In addition, we study the transmission cost (in terms of bandwidth) for the different BSs (i.e. MBS, SBS_1, SBS_3, SBS_5). Fig. 11 illustrates that the bandwidth associated with MBS was fully used, while SBSs have some bandwidth use variations, which are related to the number of accessed multicast clusters. Fig. 12 shows the variation of computing costs for MBS and three SBSs (CPU usage is shown in Table 2). The results reveal that most of the tasks for VR processing are assigned to MBS, since MBS has a longer period in a full load state compared with SBSs.

6.3 Evaluation using a System Prototype

We have built a spherical video streaming system based on the open-source framework **srs**[5]. We compare the proposed solution's service performance and resource efficiency with those of three state-of-art solutions called, D2D-assisted online reinforcement learning (RL) (**WoLF-PHC**) [7], RL-based adaptive bitrate solution (**Pensieve**) [38], and MEC-assisted joint caching and computing (**MEC-JCC**)[31].

- **WoLF-PHC** : the authors consider a time-slot system and define the system state as the relationship between users and an access point (AP) (i.e. MBS, SBSs and others). The action state is user AP selection. The system reward is associated with the throughput of overall users. According to a multi-agent reinforcement learning technique, the solution takes the users as multi-agents and lets each agent make a decision based on a joint reward function.
- **Pensieve** : the authors proposed a RL-based adaptive bitrate solution to optimize the QoE. They trained a deep neural network to make video resolution selections for live video services according to the empirical observations of the precious experiences. Pensieve supports FoV-based VR video service since its ABR algorithm can be used in conjunction with FoV prediction mechanism. We use ground-truth FoV records of dataset [43] as the predicted results.
- **MEC-JCC** : the authors formulate the joint caching and computing optimization as a multiple-choice multiple dimensional knapsack problem. The authors further design an efficient greedy algorithm by prioritizing caching and processing the VR content with the highest reward to mobile VR devices.

The prototype topology is illustrated in Fig. 13. It includes a central server (CS), a panoramic video collector,

[5] https://github.com/ossrs/srs

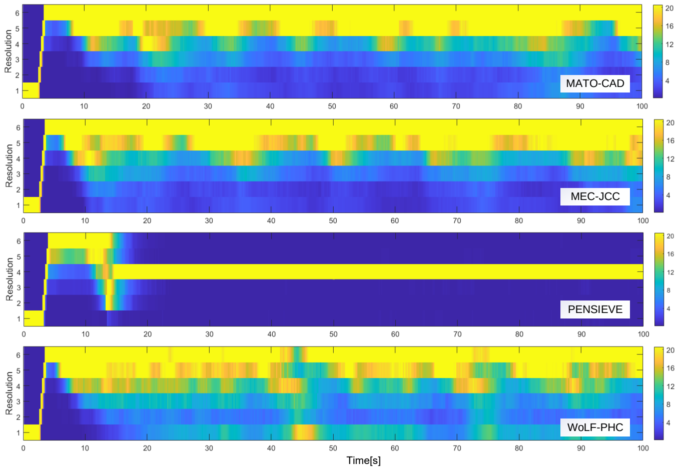


Fig. 18. The number of clients for each FoV representation vs. time.

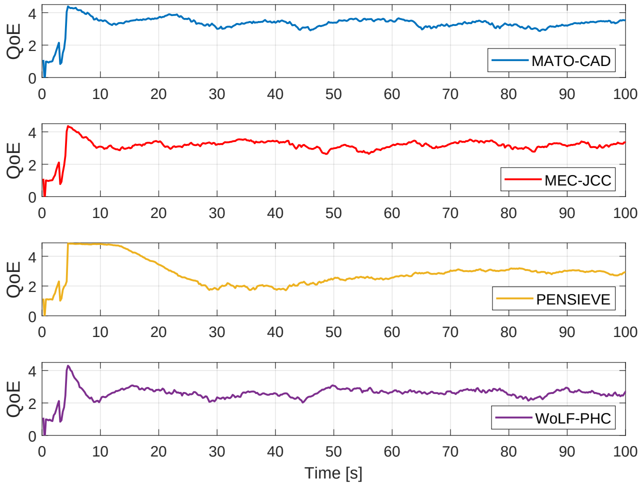


Fig. 19. Instantaneous QoE performance for the four solutions vs. time.

edge server (ES), and mobile terminals (MTs). We use two servers (Lenovo SR550) as CS. Further, we set up multiple Docker[6] containers as ES and allocate viewers two lab computers (Intel i7-7700k, Quad-Core 4.2Ghz/16GB and AMD Ryzen 5 Quad-Core 3.2GHz/16GB). All nodes in our prototype system are running on Centos 7. We use **H**TTP **L**ive **S**treaming (**HLS**) to push the VR streaming from the collector to CS through the 5G network. When MTs request live VR content, they need to obtain server status from CS and decide which network to access. ES and CS use **FFmpeg**[7] to transcode the VR tile into the demanded resolution. We set the bandwidth capacity of CS to 100Mbps as MBS and the ES to 100Mbps as SBS by configuring the uplink bandwidth. In addition, we set the maximum D2D bandwidth to 30Mbps. For convenience, we use the ground-truth FoV records as the FoV prediction results to drive prototype-based experiments.

We present prototype-based experiment results for start-up latency (SL) with different numbers of concurrent viewers in Fig. 14. The increasing trend of SL indicates that the live VR system capacity is increasingly being occupied by the live VR service with a growing number of concurrent users. However, when compared with MEC-JCC, Pensieve, and WoLF-PHC, our solution has the lowest SL, which means it supports a premium viewer playback experience. To evaluate the playback performance, we show the PFR performance of four solutions with different numbers of concurrent users in Fig. 15. Both Pensieve and WoLF-PHC provide good PFR performance when there are five concurrent users. However, the PFR performance of these two solutions sharply degraded as the number of users grew to 15 and 10, respectively. Since these two solutions are unicast, the average bandwidth usage of each user will decrease sharply with the growth of concurrent users. When the available bandwidth is not enough to support all users, severe playback freezes will occur. Therefore, our method can provide the best PFR performance in general.

Fig. 16 shows the bitrate change ratio (BCR) variation of the four tested solutions with 25 concurrent viewers. BCR curves of MATO-CAD, MEC-JCC, and WoLF-PHC go through a quick process of growth followed by a rapid decline and finally get stabilized over time. Pensieve's curve has a sharp change in the middle and then stabilizes. As time progresses, the four solutions adjust the bitrate and eventually reach a stable stage. Fig. 17 reveals the average throughput (AT) achieved by the four methods over time and can be noted how the proposed solution has significant advantages over the others. The AT performance of MATO-CAD is significantly higher than those of Pensieve and WoLF-PHC by about 50% and 18% , respectively. Although the performance of MEC-JCC is comparable to our solution in the first 80s, there is a rapid degradation for MEC-JCC in the last 20s.

In addition, we also provide details of bitrate change and a comprehensive QoE performance assessment. Fig. 18 shows the number of users that retrieve certain quality levels of each tile for the four tested solutions. Six resolutions are considered, as presented in Table 2. The brighter the color in the figure is, the higher is the number of users accessing the video with this representation. Clearly, in both MATO-CAD and MEC-JCC cases, more users can retrieve the video tiles with a higher bitrate given the brighter color in the space for resolutions 5 and 6. Besides, the large bright area in WoLF-PHC also implies there are not smooth playbacks for clients using WoLF-PHC, which confirms the BCR results illustrated in Fig. 16. In addition, although Pensieve can provide users with a stable resolution, its strategy is conservative, so the video quality accessed by users is the worst. We also provide a comparison of the four solutions in terms of QoE performance based on eq. (19) in Fig. 19. The integrated QoE scores of all the benchmark methods varies a lot before it stabilizes. However,the proposed approach has the highest stable value when compared with the other three methods and provides users with the best viewing experience in the given context.

7 Conclusion and Future Work

This paper introduces the MEC-DC framework which employs a novel buffer-based evolution model for live VR services in a 5G-HetNet and a novel buffer-based multicast scheme to support high-quality services. By considering the dynamic nature of the mobile network and time-sensitive demand, we formulate the multicast-aware task offloading (MATO) problem as a constrained optimization and devise

[6] https://www.docker.com
[7] https://ffmpeg.org

a joint optimization of data scheduling and task offloading algorithm to achieve high-quality and cost-efficient services. The paper presents both theoretical and trace-driven experimental results to demonstrate the correctness of the proposed solution and show its advantages in terms of throughput, latency, and cost consumption in comparison with other state-of-art solutions. In future work, the issue of user clustering and FoV prediction will be studied to make the solution more comprehensive. Furthermore, we will consider more generalized scenarios, including multiple macro-cells.

REFERENCES

[1] C. Yao, X. Wang, Z. Zheng, G. Sun and L. Song, "EdgeFlow: Open-Source Multi-layer Data Flow Processing in Edge Computing for 5G and Beyond," *IEEE Network*, vol. 33, no. 2, pp. 166-173, March/April 2019.

[2] Tejasvi T R, Manjaiah D H. "Energy and spectral efficient resource allocation in 5G HetNet using optimized deep bi-BRLSTM model," *Transactions on Emerging Telecommunications Technologies*, pp. e4471, 2022.

[3] F. Hu, Y. Deng, W. Saad, M. Bennis and A. H. Aghvami, "Cellular-Connected Wireless Virtual Reality: Requirements, Challenges, and Solutions," *IEEE Communications Magazine*, vol. 58, no. 5, pp. 105-111, May 2020.

[4] F. Guo, F. R. Yu, H. Zhang, H. Ji, V. C. M. Leung and X. Li, "An Adaptive Wireless Virtual Reality Framework in Future Wireless Networks: A Distributed Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8514-8528, Aug. 2020.

[5] E. Bastug, M. Bennis, M. Medard and M. Debbah, "Toward Interconnected Virtual Reality: Opportunities, Challenges, and Enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110-117, June 2017.

[6] Yang S, Hu J, Jiang K, et al. "Hybrid-360: An adaptive bitrate algorithm for tile-based 360 video streaming," *Transactions on Emerging Telecommunications Technologies*, pp. e4430, 2021.

[7] L. Feng, Z. Yang, Y. Yang, X. Que and K. Zhang, "Smart Mode Selection Using Online Reinforcement Learning for VR Broadband Broadcasting in D2D Assisted 5G HetNets," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 600-611, June 2020.

[8] S. Sukhmani, M. Sadeghi, M. Erol-Kantarci and A. El Saddik, "Edge Caching and Computing in 5G for Mobile AR/VR and Tactile Internet," *IEEE MultiMedia*, vol. 26, no. 1, pp. 21-30, 1 Jan.-March 2019.

[9] K. Long, Y. Cui, C. Ye and Z. Liu, "Optimal Wireless Streaming of Multi-Quality 360 VR Video by Exploiting Natural, Relative Smoothness-enabled and Transcoding-enabled Multicast Opportunities," *IEEE Transactions on Multimedia*, vol. 99, no. 99, pp. 1-1, Oct. 2020.

[10] Y. Liu, J. Liu, A. Argyriou and S. Ci, "MEC-Assisted Panoramic VR Video Streaming Over Millimeter Wave Mobile Networks," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1302-1316, May 2019.

[11] C. Perfecto, M. S. Elbamby, J. D. Ser and M. Bennis, "Taming the Latency in Multi-User VR 360°: A QoE-Aware Deep Learning-Aided Multicast Framework," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2491-2508, April 2020.

[12] X. Feng, Y. Liu and S. Wei, "LiveDeep: Online Viewport Prediction for Live Virtual Reality Streaming Using Lifelong Deep Learning," *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, Atlanta, GA, USA, 2020, pp. 800-808.

[13] X. Feng, Z. Bao and S. Wei, "LiveObj: Object Semantics-based Viewport Prediction for Live Mobile Virtual Reality Streaming," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2736-2745, May 2021.

[14] M. Chen, W. Saad, C. Yin and M. Debbah, "Data Correlation-Aware Resource Management in Wireless Virtual Reality (VR): An Echo State Transfer Learning Approach," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4267-4280, June 2019.

[15] A. A. Simiscuka, T. M. Markande and G. Muntean, "Real-Virtual World Device Synchronization in a Cloud-Enabled Social Virtual Reality IoT Network," *IEEE Access*, vol. 7, pp. 106588-106599, Aug.2019.

[16] Z. He, L. You, R. W. Liu, F. Yang, J. Ma and N. Xiong, "A Cloud-Based Real Time Polluted Gas Spread Simulation Approach on Virtual Reality Networking," *IEEE Access*, vol. 7, pp. 22532-22540, Jan.2019.

[17] J. Yang, J. Luo, J. Wang and S. Guo, "CMU-VP: Cooperative Multicast and Unicast With Viewport Prediction for VR Video Streaming in 5G H-CRAN," *IEEE Access*, vol. 7, pp. 134187-134197, Sept.2019.

[18] Y. Li, C. Hsu, Y. Lin, and C. Hsu, "Performance Measurements on a Cloud VR Gaming Platform," *Proceedings of the 1st Workshop on Quality of Experience (QoE) in Visual Multimedia Applications*, Seattle, WA, USA, pp. 37-45, 2020.

[19] J. Dai, Z. Zhang, S. Mao and D. Liu, "A View Synthesis-Based 360° VR Caching System Over MEC-Enabled C-RAN," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3843-3855, Oct. 2020.

[20] J. Chakareski, "Viewport-Adaptive Scalable Multi-User Virtual Reality Mobile-Edge Streaming," *IEEE Transactions on Image Processing*, vol. 29, pp. 6330-6342, May 2020.

[21] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang and X. Chu, "MEC-Assisted Immersive VR Video Streaming Over Terahertz Wireless Networks: A Deep Reinforcement Learning Approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9517-9529, Oct. 2020.

[22] Y. Zhang, L. Jiao, J. Yan and X. Lin, "Dynamic Service Placement for Virtual Reality Group Gaming on Mobile Edge Cloudlets," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1881-1897, Aug. 2019.

[23] L. Hu, Y. Tian, J. Yang, T. Taleb, L. Xiang and Y. Hao, "Ready Player One: UAV-Clustering-Based Multi-Task Offloading for Vehicular VR/AR Gaming," *IEEE Network*, vol. 33, no. 3, pp. 42-48, May/June 2019.

[24] J. Chakareski and S. Gupta, "Multi-Connectivity and Edge Computing for Ultra-Low-Latency Lifelike Virtual Reality," *2020 IEEE International Conference on Multimedia and Expo (ICME)*, London, United Kingdom,

2020, pp. 1-6.

[25] Y. Zhou, L. Tian, L. Liu and Y. Qi, "Fog Computing Enabled Future Mobile Communication Networks: A Convergence of Communication and Computing," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 20-27, May 2019.

[26] E. Markakis, D. Negru, J. Bruneau-Queyreix, et al. "A p2p home-box overlay for efficient content distribution," *Emerging Innovations in Wireless Networks and Broadband Technologies*, IGI Global, pp. 199-220, 2016.

[27] C. Xu et al., "The Case for FPGA-based Edge Computing," *IEEE Transactions on Mobile Computing*, vol. 99, no. 99, pp. 1-1, Dec 2020.

[28] K. Karras, E. Pallis , G. Mastorakis, Y. Nikoloudakis, J.Batalla, C. Mavromoustakis, E. Markakis. "A hardware acceleration platform for AI-based inference at the edge," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 1059-1070, 2020.

[29] Y. Zhou, C. Pan, P. L. Yeoh, K. Wang, M. Elkashlan, B. Vucetic and Y. Li, "Communication-and-Computing Latency Minimization for UAV-Enabled Virtual Reality Delivery Systems," *IEEE Transactions on Communications*, vol. 69, no. 3, pp. 1723-1735, March 2021.

[30] T. Dang and M. Peng, "Joint Radio Communication, Caching, and Computing Design for Mobile Virtual Reality Delivery in Fog Radio Access Networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 7, pp. 1594-1607, July 2019.

[31] Y. Sun, Z. Chen, M. Tao and H. Liu, "Communications, Caching, and Computing for Mobile Virtual Reality: Modeling and Tradeoff," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7573-7586, Nov. 2019.

[32] Cui, E, Yang, D, Wang, H, Zhang, W. "Learning-based deep neural network inference task offloading in multi-device and multi-server collaborative edge computing," *Transactions on Emerging Telecommunications Technologies*, pp. e4485, 2022.

[33] X. Chen, C. Xu, M. Wang, Z. Wu, L. Zhong and L. A. Grieco, "Augmented Queue-based Transmission and Transcoding Optimization for Livecast Services Based on Cloud-Edge-Crowd Integration," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4470-4484, Nov. 2021.

[34] X. Chen, C. Xu, M. Wang, Z. Wu, S. Yang, L. Zhong and G.-M. Muntean, "A Universal Transcoding and Transmission Method for Livecast with Networked Multi-Agent Reinforcement Learning," *Proceedings of the 40th IEEE Conference on Computer Communications (IEEE INFOCOM 2021)*, Vancouver BC Canada, pp. 1-10, 2021.

[35] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh and E. Modiano, "Scheduling Policies for Minimizing Age of Information in Broadcast Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2637-2650, Dec. 2018.

[36] Y. Hsu, E. Modiano and L. Duan, "Scheduling Algorithms for Minimizing Age of Information in Wireless Broadcast Networks with Random Arrivals," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2903-2915, Dec. 2020.

[37] I. Kadota and E. Modiano, "Minimizing the Age of Information in Wireless Networks with Stochastic Arrivals," *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '19)*, pp. 221-230, New York, NY, USA, 2019.

[38] H. Mao, R. Netravali, and M. Alizadeh. "Neural Adaptive Video Streaming with Pensieve," *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (*SIGCOMM '17*). Association for Computing Machinery, New York, NY, USA, 197–210, 2017.

[39] C. Guo, Y. Cui and Z. Liu, "Optimal Multicast of Tiled 360 VR Video in OFDMA Systems," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2563-2566, Dec. 2018.

[40] H. Ahmadi, O. Eltobgy, and M. Hefeeda, "Adaptive Multicast Streaming of Virtual Reality Content to Mobile Users," *Proceedings of the on Thematic Workshops of ACM Multimedia*, pp. 170-178, 2017.

[41] C. Guo, Y. Cui and Z. Liu, "Optimal Multicast of Tiled 360 VR Video," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 145-148, Feb. 2019.

[42] Y. Bao, T. Zhang, A. Pande, H. Wu and X. Liu, "Motion-Prediction-Based Multicast for 360-Degree Video Transmissions," *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, San Diego, CA, 2017, pp. 1-9.

[43] W. Lo, C. Fan, J. Lee, C. Huang, K. Chen, and C. Hsu, "360° Video Viewing Dataset in Head-Mounted Virtual Reality," *Proceedings of the 8th ACM on Multimedia Systems Conference*, Taipei, Taiwan, pp. 211-216, 2017.

[44] H. Nam, K. Kim and H. Schulzrinne, "QoE matters more than QoS: Why people stop watching cat videos," *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1-9.

[45] B. Rainer, D. Posch and H. Hellwagner, "Investigating the Performance of Pull-Based Dynamic Adaptive Streaming in NDN," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2130-2140, Aug. 2016.

[46] B. Zhou, Y. Cui and M. Tao, "Stochastic Content-Centric Multicast Scheduling for Cache-Enabled Heterogeneous Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6284-6297, Sept. 2016.

[47] C. Xu, M. Wang, X. Chen, L. Zhong and L. A. Grieco, "Optimal Information Centric Caching in 5G Device-to-Device Communications," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2114-2126, Sept. 2018.

[48] X. Chen, C. Xu, M. Wang, T. Cao, L. Zhong and G. Muntean, "Optimal Coded Caching in 5G Information-Centric Device-to-Device Communications," *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1-7.

[49] Menkovski V. "Computational inference and control of quality in multimedia services," *Springer*, 2015.

[50] B. Rainer, S. Lederer, C. Müller and C. Timmerer, "A seamless Web integration of adaptive HTTP streaming," *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pp. 1519-1523, 2012.

[51] Y. Xu, G. Gui, H. Gacanin and F. Adachi, "A Survey on Resource Allocation for 5G Heterogeneous Networks:

Current Research, Future Trends, and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 668-695, Second quarter 2021.

[52] M. J. Neely, "Stochastic Network Optimization with Application to Communication and Queueing Systems," *Morgan & Claypool*, vol. 3, no. 1, pp. 1-211, 2010.

[53] H. Pang, C. Zhang, F. Wang, H. Hu, Z. Wang, J. Liu, and L. Sun, "Optimizing Personalized Interaction Experience in Crowd-Interactive Livecast: A Cloud-Edge Approach," *ACM Conference on Multimedia (MM '18)*, Seoul, Republic of Korea, pp. 1217-1225, 2018.

[54] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A Dataset for Exploring User Behaviors in VR Spherical Video Streaming," *Proceedings of the 8th ACM on Multimedia Systems Conference*, Taipei, Taiwan, pp. 193-198, 2017.

[55] L. Yu, T. Tillo and J. Xiao, "QoE-Driven Dynamic Adaptive Video Streaming Strategy With Future Information," *IEEE Transactions on Broadcasting*, vol. 63, no. 3, pp. 523-534, Sept. 2017.

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, China. She has published papers in prestigious international journals and conferences in the related area, including IEEE COMMUNICATION MAGAZINE, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE INTERNET THINGS JOURNAL, IEEE INFOCOM and ACM MULTIMEDIA, etc. Her research interests include communication networks, computer system and architecture, and mobile Internet technology.

Xingyan Chen received the Ph. D degree in computer technology from Beijing University of Posts and Telecommunications (BUPT), in 2021. He is currently a lecturer with the School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu. He has published papers in well-archived international journals and proceedings, such as the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and IEEE INFOCOM etc. His research interests include Multimedia Communications, Multi-agent Reinforcement Learning and Stochastic Optimization.

Changqiao Xu [SM'15] received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences (ISCAS) in Jan. 2009. He was an Assistant Research Fellow and R&D Project Manager in ISCAS from 2002 to 2007. He was a researcher at Athlone Institute of Technology and Joint Training PhD at Dublin City University, Ireland during 2007-2009. He joined Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in Dec. 2009. Currently, he is a Professor with the State Key Laboratory of Networking and Switching Technology, and Director of the Network Architecture Research Center at BUPT. His research interests include Future Internet Technology, Mobile Networking, Multimedia Communications, and Network Security. He has edited two books and published over 200 technical papers in prestigious international journals and conferences, including IEEE/ACM ToN, IEEE TMC, IEEE INFOCOM, ACM Multimedia etc. He has served a number of international conferences and workshops as a Co-Chair and TPC member. He is currently serving as the Editor-in-Chief of TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES (WILEY). He is Senior member of IEEE.

Yunxiao Ma received the B.E. degree in telecommunications engineering from the School of Electronic Information Engineering, Inner Mongolia University, in 2019. She is currently pursuing the Ph.D. degree with the Network Architecture Research Center, School of Computing, Beijing University of Posts and Telecommunications, under the supervision of Prof. Changqiao Xu. Her research interests include multimedia communications and stochastic optimization.

Mu Wang received his M.S. and a Ph.D. degree in computer technology from Beijing University of Posts and Telecommunications (BUPT) in 2015 and 2020. He was a Joint Ph.D. student at the School of Electrical, Computer, and Energy Engineering (ECEE), Arizona State University. He is currently a postdoctoral research associate with the Department of Computer Science and Technology & BNRist, Tsinghua University. His research interests include information-centric networking, wireless communications, and multimedia sharing over wireless networks.

Yu Zhao received the B.S. degree from Southwest Jiaotong University in 2006, and the M.S. and Ph.D. degrees from the Beijing University of Posts and Telecommunications in 2011 and 2017, respectively. He is currently an Associate Professor at Southwestern University of Finance and Economics. His current research interests include natural language processing, knowledge graph, machine learning, and recommendation system.

Gabriel-Miro Muntean [SM 17] is a Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and coDirector of DCU Performance Engineering Laboratory. He has published 4 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.

APPENDIX A

According to the buffer evolution model, we have the sum of equations (2) for all t as follows:

$$\chi_u[T] - \chi_u[0] = \sum_{t=0}^{T-1} (\tau - \chi_u[t])\gamma_u[t]h_u[t] - T \tag{19}$$

As the initial buffer $\chi_u[0] = 0$, we have:

$$\frac{1}{T}\chi_u[T] = \frac{1}{T}\sum_{t=0}^{T-1} (\tau - \chi_u[t])\gamma_u[t]h_u[t] - 1 \tag{20}$$

Because $\gamma_u \in \Psi$, we have $\liminf_{T\to\infty} \frac{1}{T}\mathbb{E}[\chi_u[T]] = 0$. Take the expectation of both sides of eq. (20) and we get the following equation for $T \to \infty$.

$$\begin{aligned}\liminf_{T\to\infty}\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} \chi_u[t]\gamma_u[t]h_u[t]\right] = \\ \tau \limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} \gamma_u[t]h_u[t]\right] - 1\end{aligned} \tag{21}$$

Divide $\limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} \gamma_u[t]h_u[t]\right]$ for both sides of eq. (21), we have **Lemma 1**.

$$\bar{\chi}_u[t] = \tau - \frac{1}{\limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} \gamma_u[t]h_u[t]\right]} \tag{22}$$

APPENDIX B

Combining eq. (7) and eq. (10a), for $\forall s_n \in \mathbb{S}$ at time t, we have:

$$\begin{aligned}\mathcal{J}(\boldsymbol{\pi}[t]) = \sum_{u\in\Omega} \Big(\lambda_u \pi_u[t] + \\ \gamma_u[t]\big(\alpha_u p_n(\pi_u[t]) + \beta_u p_r(\pi_u[t])\big)\Big)\end{aligned} \tag{23a}$$

$$s.t. \sum_{u\in\Omega_{n^\dagger}} \gamma_u \alpha_u[t] \le b_n, \quad \sum_{u\in\Omega_{n^\dagger}} \lambda_u \pi_u[t] \le c_n \tag{23b}$$

Based on eq. (8) and eq. (23a), when the transcoding task is deployed in the n-th BS, we get the expected cost of the n-th BS: $(\lambda_u + \alpha_u(1-p_0))$ at t. Otherwise, the cost is $\beta_u(1-p_0)$. According to the definition ψ_u, we have **Lemma 2**.

APPENDIX C

We define $L[t] = \frac{1}{2}\sum_{u\in\Omega} q_u^2[t]$ and $\Delta[t] = L[t+1] - L[t]$. According to virtual queue update in eq. (15), we have:

$$\begin{aligned} q_u^2[t+1] &= \Big[max\{q_u[t] + \sqrt{V/q_u[t]} - \gamma_u[t]h_u[t], 1\}\Big]^2 \\ &\le 1 + \big(q_u[t] + \sqrt{V/q_u[t]} - \gamma_u[t]h_u[t]\big)^2 \\ &= 1 + q_u^2[t] + \big(\sqrt{V/q_u[t]} - \gamma_u[t]h_u[t]\big)^2 \\ &\qquad + 2q_u[t]\big(\sqrt{V/q_u[t]} - \gamma_u[t]h_u[t]\big)\end{aligned}$$

Because $\gamma_u[t]h_u[t] \in \{0,1\}$ and $q_u[t] \ge 1$, we have:

$$\Delta[t] \le \frac{V+1}{2}|\Omega| + \sum_{u\in\Omega} q_u[t]\big(\sqrt{V/q_u[t]} - \gamma_u[t]h_u[t]\big) \tag{24}$$

We denote the objective function from eq. (12) as $f(\varepsilon_u) = \sum_{u\in\Omega} \frac{V}{\varepsilon_u}$ and we have:

$$\begin{aligned} Vf(\varepsilon_u) + \Delta[t] \le \sum_{u\in\Omega} \frac{V}{\varepsilon_u} + (1 + \frac{V}{2})|\Omega| \\ + \sum_{u\in\Omega} q_u[t]\big(\sqrt{V/q_u[t]} - \gamma_u[t]h_u[t]\big)\end{aligned} \tag{25}$$

Summing both sides of eq. (25) over the entire time, dividing both sides by T and then taking the expected value, we have:

$$\begin{aligned}\mathbb{E}\Big[\frac{V}{T}\sum_{t=0}^{T-1} f(\boldsymbol{\varepsilon}[t])\Big] + \mathbb{E}[L[T] - L[0]] \le \mathbb{E}\Big[\frac{1}{T}\sum_{t=0}^{T-1}\sum_{u\in\Omega}\frac{V}{\varepsilon_u[t]}\Big] \\ + \frac{V+2}{2}|\Omega| + \sum_{u\in\Omega}\mathbb{E}\Big[\frac{1}{T}\sum_{t=0}^{T-1} q_u[t]\Big(\sqrt{V/q_u[t]} - \gamma_u[t]h_u[t]\Big)\Big]\end{aligned} \tag{26}$$

According to eq. (4) and eq. (15), we have $\lim_{T\to\infty}\mathbb{E}\big[\frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\frac{V}{q_u[t]}}\big] \le \lim_{T\to\infty}\mathbb{E}\big[\frac{\sum_{t=0}^{T-1}\gamma_u[t]h_u[t]}{T}\big]$. Additionally $L[t] \ge 0$, we use $\sqrt{V/q_u[t]}$ to substitute $\varepsilon_u[t]$ and get:

$$\begin{aligned}\mathbb{E}\Big[\frac{V}{T}\sum_{t=0}^{T-1} f(\boldsymbol{\varepsilon}[t])\Big] \le \mathbb{E}\Big[\frac{2}{T}\sum_{t=0}^{T-1}\sum_{u\in\Omega}\sqrt{Vq_u[t]}\Big] + \frac{V+1}{2}|\Omega| \\ - \sum_{u\in\Omega}\mathbb{E}\Big[\frac{1}{T}\sum_{t=0}^{T-1} q_u[t]\gamma_u[t]h_u[t]\Big] + \mathbb{E}[L[0]]\end{aligned} \tag{27}$$

Replacing $\gamma_u[t]$ with $\gamma_u^*[t]$. As $\varepsilon_u[t] = \mathbb{E}[\gamma_u[t]h_u[t]]$, we have:

$$\begin{aligned}\mathbb{E}\Big[\frac{V}{T}\sum_{t=0}^{T-1} f(\boldsymbol{\varepsilon}[t])\Big] \le \mathbb{E}\Big[\frac{V}{T}\sum_{t=0}^{T-1} f(\varepsilon_u^*[t])\Big] + \frac{V+1}{2}|\Omega| \\ - \sum_{u\in\Omega}\mathbb{E}\Big[\frac{1}{T}\sum_{t=0}^{T-1}\varepsilon_u^*[t]\Big[\sqrt{q_u[t]} - \frac{\sqrt{V}}{\varepsilon_u^*[t]}\Big]^2\Big] + \mathbb{E}[L[0]]\end{aligned} \tag{28}$$

where the $\varepsilon_u^*[t]$ is the optimal solution to the problem from eq. (12). Since the last term of eq. (28) is always greater than 0, we get:

$$\mathbb{E}\Big[\frac{1}{T}\sum_{t=0}^{T-1} f(\boldsymbol{\varepsilon}[t])\Big] \le \mathbb{E}\Big[\frac{1}{T}\sum_{t=0}^{T-1} f(\varepsilon_u^*[t])\Big] + \frac{V+1}{2V}|\Omega| \tag{29}$$

Combining **Lemma 1** and eq. (29), we proof **Theorem 1**.