A Transcoding-enabled 360° VR Video Caching and Delivery Framework for Edge-enhanced Next-generation Wireless Networks

Han Xiao, Changqiao Xu, *Senior Member, IEEE,* Zichen Feng, Renjie Ding, Shujie Yang, Lujie Zhong, Jie Liang and Gabriel-Miro Muntean *Senior Member, IEEE,*

***Abstract*—Virtual reality (VR) content, including 360 panoramic video, provides users with an immersive multimedia experience and therefore attracts increasing research and development attention. However, the requirement of high bandwidth and low latency of virtual reality service demand puts forward greater challenges to the current infrastructure, especially mobile networks. Inspired by the sharable nature of virtual reality content tiles, we further considered the potential opportunities for computing, caching, and multicast to address the challenges of transmission of panoramic content. This paper proposes a novel transcoding-enabled VR video caching and delivery framework for edge-enhanced next-generation wireless networks. Firstly, an edge cooperative caching scheme based on multi-agent reinforcement learning is introduced to improve the utilization efficiency of computing and storage resources, and then reduce service delay. Second, a two-tier NOMA-based base station-multicast group matching mechanism is designed to solve the collaboration challenge during the edge delivery process. A series of experiments have demonstrated the advantages of the proposed scheme in terms of cache hit rate, latency and other aspects in comparison with alternative approaches.**

***Index Terms*—Virtual Reality, Multicast Delivery, Cooperative Cache, Deep Reinforcement Learning**

I. Introduction and Motivation

Benefiting from the support of 5G, virtual reality (VR) has been deemed as a representative use case of the next generation applications, aiming at further improving the daily life of human beings. Unlike traditional flat video, VR video allows users to enter the scene as a participant. This enables users to gain a strong sense of immersion and enjoy a rich detailed experience. It is therefore no surprise that, according to a Cisco research report [1], VR headset market is expected to grow five-fold, from $18 million in 2016 to nearly $100 million in 2021.

However, the VR content characteristics place higher requirements on the next-generation mobile networks and beyond (i.e. 5G/B5G). On one hand, the panoramic VR video requires delivery of multiple screens-worth data in order to cope with the dynamic viewpoint change, which tremendously increases the transmission overhead. For example, the bitrate of a VR video (400Mbps) is larger than that of a 2-D video (30Mbps) by an order of magnitude [2]–[4]. On the other hand, ultra-low latency ($\leq$ 30ms) is required to avoid viewer motion sickness. Supporting both high bitrate and low latency simultaneously hinders the wide use of VR content transmissions and is an important issue to be solved urgently.

Naturally, edge caching is considered as a viable solution to alleviate this situation. The popular content is cached in proximity of the viewers at the network edge, generally at base stations (BS). The redundant duplicated traffic on the backhaul link is significantly reduced and the service delay is lowered. However, compared to traditional video caching, some issues need to be further explored.

The first issue is what content needs to be cached. Caching a whole VR video is both impractical and unnecessary. Based on the VR bitrate mentioned above, a five-min video requires a storage space of 15GB [2]. Even a solution with a BS backed by a powerful edge server would not be scalable. At the same time, typically, a head-mounted display (HMD) is equipped with a screen with a field of view (FoV) with limited height and less than a third of the 360 degree range[1]. It means that the content viewed takes up only a small portion (about $12-20\%$ [5]) of the VR frame. Caching integrally the VR content is essentially a waste of bandwidth and storage space. Instead, each panoramic video frame is usually projected and involves a transformation from an omnidirectional view to a 2-D plane. Furthermore, the video is tiled and a FoV-based adaptation transmission is employed. However, the direction of view is dynamic and there is a need to carefully determine which tiles are worth caching.

A second important issue is coping with demands for content of various bitrates. Depending on the buffer state of the client and network conditions, the requirements of viewers for bitrate vary greatly. In general, the multi-bitrate video encoding using DASH is adopted to provide viewers with

Manuscript received August 15, 2021; revised November 15, 2021; accepted December 17, 2021. This work is supported by the National Natural Science Foundation of China (NSFC) under grant No. 61871048, 61872253, and 62001057 and by the 111 Project (B18008). G.-M. Muntean wishes to acknowledge the Science Foundation Ireland (SFI)'s support via grant nos. 16/SP/3804 (Enable) and 12/RC/2289_P2 (Insight). (*Corresponding author: Changqiao Xu.*)

H. Xiao, C. Xu, Z. Feng, R. Ding and S. Yang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China. E-mail: {xiaohan, cqxu, fengzichen1027, 2017213116, sjyang}@bupt.edu.cn

L. Zhong is with the Information Engineering College, Capital Normal University, Beijing 100048, China. E-mail: zhonglj@cnu.edu.cn.

J. Liang is with the Research Institute of China Telecom, Ave Zhongshan, Guangzhou 510630, China. E-mail: liangjie6@chinatelecom.cn.

G.-M. Muntean is with the Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Dublin 9, Ireland. E-mail: gabriel.muntean@dcu.ie.

[1]The FoV of a HTC Vive is 110° and that of an Oculus Quest is 100°.

diverse video versions and enable adaptive delivery flexibility [6], [7]. This results in an increase complexity of the cache decisions. Caching is not limited to the content selection, but also needs to accommodate changes in the bitrate. The high-bitrate tiles can be converted to low-bitrate tiles provided there are enough computational resources. However, which encoded video versions should be cached remains an important factor which influences the user experience. In this context, the transcoding-enabled cache provides support for flexible increased quality video services. In difficult network delivery conditions, transcoding converts high bitrate content into low bitrate content, avoiding emptying of the viewer buffer and freezing of the display, which would negatively affect viewers quality of experience. In such conditions, a lower content bitrate is selected, less information is transmitted, which eventually goes through the congested network, filling the video buffer and ensuring the smoothness of video playback. This continuous playback of video, although at slightly lower bitrate, is much preferred to one interrupted by buffering and results in increased user experience [8]. Transcoding[2] is used to complement caching when the high quality content is not available and not to replace it. When a high bitrate content request comes in and the cache has it, it will serve it, supporting high quality of service.

In addition, edge delivery is a primary bottleneck during VR content service due to the limited communication resource availability. Benefiting from the fact that shared tiles exist in the FoV of multiple viewers[3], multicast can be employed to support VR service delivery. By multicast delivering the shared tiles to multiple viewers, the spectrum efficiency is improved while also reducing the average latency of edge delivery. Among the existing solutions, NOMA-based edge multicast [10] has recently attracted researchers' attention due to its positive performance. Through power domain division multiplexing (PDM), the edge BSs can transmit superposed messages in the same frequency band, satisfying the requests of multiple multicast groups. This is regarded as an effective advanced solution for content delivery in the next-generation wireless networks, especially in the presence of both macro BSs (MBS) and small BSs (SBS).

However, the amount of superposed content is limited and therefore the content to be multicast should be well determined. At the same time, the coverage of MBSs and SBSs differs greately[4]. Therefore, realization of a MBS-SBS collaborative multicast is an important problem and needs to be considered.

Summarizing, on one hand, the tiled virtual reality content presents a fragmented state, and the heterogeneous availability of resources such as computing and storage determines a multi-dimensional and multi-scale state space, which increases the difficulty of caching decision-making and transmission scheduling. On the other hand, the edge BSs already enable transmission of a large amounts of data and processing of user requests and can further help, including by hosting data-driven agents to be trained and employed in enhanced data delivery. In addition, the natural intersection of multiple viewers' FoVs provide great multicast opportunities. In this context, this paper proposes **a novel transcoding-enabled VR video caching and delivery framework in edge-enhanced next-generation wireless networks** and its major contributions are summarized as follows:

1) *An innovative edge service framework* which integrates caching and multicast delivery is introduced in order to achieve service latency minimization. The framework jointly considers collaboration of edge BS, dynamic aspect of network and time variability of requests, decomposing the problem into two subproblems: caching (delivery preparation phase) and delivery (execution phase).
2) Transcoding is included in the cache sub-problem, enabling the cached content to be converted to a lower bitrate and serve more viewers. Further, the collaborative caching of edge MBS and SBS is formulated as a networked multi-agent Markov decision process [11]. The concept of team-average-reward is introduced to describe the global cache performance at the edge. *A multi-agent Actor-Critic algorithm* is proposed to minimize the latency of delivery preparation.
3) *A two-tier BS-multicast group matching algorithm* to support collaborative MBS and SBS multicast content delivery is designed. The stability and Pareto optimality of the matching are proved and the computational complexity of the solution is analyzed.
4) A series of experiments conducted have verified the effectiveness of the proposed approach on a testbed based on the *Aframe Player* [12]. The results demonstrate that the proposal outperforms alternative solutions in terms of hit rate, time delay and multicast coverage ratio, etc.

This paper is organized as follows. Section II discusses related works. Section III introduces the system model and section IV formalizes the problem. Section V designs the transcoding-enabled cache scheme and section VI proposes the multicast mechanism. Section VII presents and discusses the simulation results and section VIII draws conclusions.

II. Related Works

VR content delivery has attracted the attention of many relevant researchers, due to its wide application. This section gives an overview of the related works and explain how the work presented in this paper is different.

A. Edge Caching

Edge caching has been widely studied in relation to traditional multimedia due to the benefits of proximity delivery [13]–[17]. Lately the scope of its application has been expanded to cover VR [18]–[22]. Georgios *et al.* [18] studied the case of cached tile streaming encoded by DASH or SVC, respectively. They minimized the error between requested and cached content under viewport, and further transformed it into a K-Medoids problem in the data cluster. The work shows how

[2]The conversion from low bitrate to high bitrate can be achieved by employing for instance the super resolution technology [9]. This is out of the scope of this paper.

[3]Common scenarios include cinemas, shopping malls, airports, etc.

[4]MBSs cover a wide area and can multicast content to more users, while SBSs cover a smaller area.

to cope with the various bitrate requirements under different coding rules. A FOV-aware cache strategy was proposed in [19] and a Naive-Bayes-based scheme was adopted to learn the probabilistic model of tile access, so as to determine the priority of the cache replacement when the storage space overflows. Machine learning is introduced in this scheme, which improves the adaptability of nodes to the dynamic environment to some extent. Maniotis $et\ al.$ [20] introduced the innovative concept of virtual viewport. The cache decision was simplified to the cached video and the cached virtual viewport. A DQN-based method was designed to determine the optimal placement strategy of the cache, so as to maximize the quality of the delivered video. Benefiting from the powerful fitting ability of a neural network, an agent is designed to cope with the dynamic and complex communication environment more flexibly. The joint optimization of transcoding and caching was considered. Jacob $et\ al.$ [21] proved that the problem is NP-hard and designed an improved dynamic programming scheme for solving the problem within polynomial time. Zhou $et\ al.$ [22] proposed a joint optimization scheme of cache and transcoding. The solution was more targeted to the heterogeneous network of ICN and MEC, and lacks sufficient reference value for cellular networks. We also studied the problem of transcoding and transmission of multimedia content, and proposed a solution based on augmented graph [23], [24]. However, the scheme mainly solved the demand of livecast of traditional media, and cannot be directly applied to VR panoramic video scenes.

B. Multicast Delivery

In addition, multicast has been employed by researchers as an effective tool for supporting VR delivery. For example, based on the correlation between FOV and location of viewers, Perfecto $et\ al.$ [25] proposed a mmWave physical layer multicast scheme. It decoupled the multicast process into two sub-problems, i.e. request access and scheduling that can be solved independently. Long $et\ al.$ [26] considered two kinds of quality requirements (i.e. absolute smoothness and relative smoothness) and multicast modes (i.e. without transcoding and with transcoding). The opportunity of multicast was extended in comparison with original natural multicast and the resource utilization was improved. This motivated us to think whether the cache strategy with transcoding can further optimize the efficiency and performance of multicast. At the same time, TDMA-based multicast content delivery from one server to multiple users was studied in literature [27]. By optimizing transmission power and latency, the quality of received video was maximized. Tan $et\ al.$ [28] presented an energy efficient resource allocation scheme with non-orthogonal multicast and unicast transmissions. The available data rate was given and the energy efficiency was optimized. The performance of multicast millimeter-wave wireless networks was studied from the perspective of stochastic geometry and a cooperative NOMA-based multicast scheme was proposed in literature [29]. Finally, stochastic optimization of edge multicast was focused on in our previous work [30].

There is a definite benefit of using NOMA in VR transmission. The emerging multimedia content imposes higher capacity requirements to the network. Specifically, when multiple viewers request video content at the same time (i.e. in cinemas, shopping malls, etc.), the FoVs of viewers have some areas which are different, but also many areas that overlap. This means a lot of content has to be delivered repeatedly, results in unnecessary use of the limited bandwidth resources and adds to the existing pressure on delivery. Employing NOMA-based multicast can provide support for such VR transmission, making NOMA attract the attention of researchers.

For example, in literature [31], the author studied cooperative and non-cooperative NOMA-based transmission scheme design in the context of VR livecast. By deriving outage probability and designing an innovative solution for power allocation optimization, the average outage capacity was significantly increased, and the quality of user experience was improved. In [32], the authors combined NOMA with mobile edge computing and designed a dynamic computing offloading scheme to reduce the computational cost. The proposed solution considered the computing requirements of target recognition, human pose estimation and visual tracking in a VR context. The authors of [33] focused on the concurrent support of vision and touch in wireless cellular networks. For downlink transmission optimization, the authors deduced the closed-form average rate under OMA and NOMA, respectively, and concluded that the latter has better performance than the former. Additionally, other researchers stated that NOMA technology has an important impact on advanced multimedia applications (e.g. VR) in terms of improving wireless capacity [34] and resource optimization efficiency [35].

Unlike existing works, this paper introduces a combined transcoding-enabled VR video caching and delivery framework for edge-enhanced next-generation wireless networks, proposed in order to increase the quality of the delivered VR content. The framework includes a novel adaptive content delivery scheme which uses video tiles, employs innovative FoV content selection and relies on multicast to improve the VR content delivery performance. The former provides a novel cooperative caching scheme to improve the probability that request can be satisfied at the edge. The latter is applied to obtain tractable solutions to combinatorial problems in a distributed manner. The stability and Pareto optimality of the proposed scheme are demonstrated in order to show its effectiveness. In addition to the differences from other works, the testbed used to implement a real VR transmission process is also an interesting contribution.

III. System Model

The scenario considered in this paper is illustrated in Fig. 1. The remote server stores all the created VR videos and transmits the content to the edge via backhaul links to serve users. An edge network is considered as a heterogeneous network, consisting of a MBS and one small cell tier. MBS and SBSs are peer nodes during the caching process and MBS acts as a regional controller during multicast. Similar to [37] and [38], the location of edge BSs and viewers is modeled as an independent homogeneous Poisson point process (PPP) with the density λ_B and λ_U, respectively. In this section, we introduce the system model involved in this paper, which includes network, content and cache models.

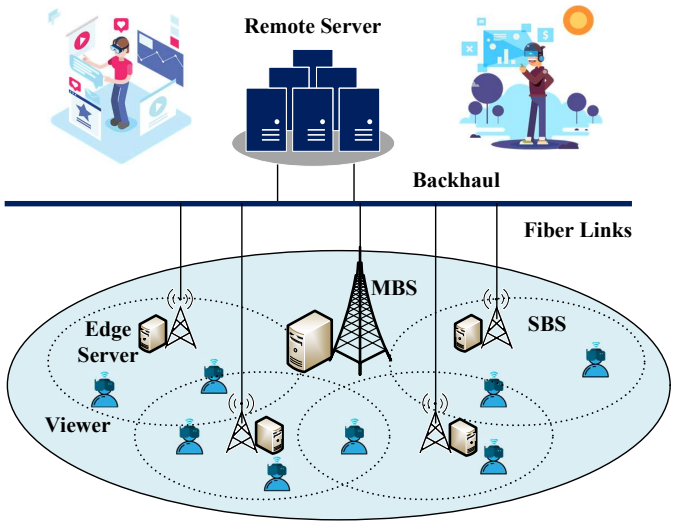


Fig. 1. VR service scenario considered in this paper. The remote server and edge BSs are connected to each other by optical fiber links. Edge heterogeneous networks provides services to viewers via wireless links.

Notably, we use lowercase *italic* symbols as scalars, e.g u, b. The lowercase *italic* **bold** symbol is vector. The calligraphic symbols are sets, e.g. $\mathcal{U}, \mathcal{B}$. The norm $|\cdot|$ denotes the size of a set/vector. The mathematical notations commonly used are summarized in Table 1.

A. Network Model

The main network elements include a remote server, edge BSs and viewers (users). The remote server collects rich VR video content. The set of viewers is represented as $\mathcal{U} = \{1, 2, ...u, ..., U\}$. The set of BSs is denoted as $\mathcal{B} = \{1, 2, ...b, ..., B\}$, where $\mathcal{B}_1$ is the macro BS (MBS) and others are the small BS (SBS). Each BS is equipped with an edge server, which endows the edge BSs with the ability of computation and storage. The BS bound to the user u is represented as $\mathcal{B}_u$. It is worth noting that the fiber links are established between BSs for the convenience of exchanging the data, which enables the BSs to communicate with each other and realize collaborative caching. BSs provide services to viewers over the wireless link.

B. Content Model

All VR video content is recorded and processed in advance. Then, it is stored in the remote server and can be freely obtained by any viewer u. In order to simplify the presentation, the content requested by the viewer is represented as the v-th video in the video library, and we omit the subscript when it is not necessary. Due to the spherical feature of the video content, it is projected to a two-dimensional plane[5]. For the convenience of transmitting, the panoramic video is encoded into G sequential Group of Pictures (GoP)[6], denoted as $\mathcal{G} = \{1, 2, ..., g, ...G\}$. Each GoP is further encoded with $|\mathcal{D}|$ different versions, $\mathcal{D} = \{1, 2, ..., d, ..., D\}$, following the DASH standard to suit various demands from viewers[7]. To be specific, it is further cropped as the tiles with equal size. The set of tiles is denoted as $\mathcal{Z}_g = \{1, 2, ..., z, ..., Z\}$ (the z-th tile of the g-th GoP is denoted as $\mathcal{Z}_{g,z}$).

Note that the higher the VR content bitrate, the better the service is for the viewers. Correspondingly, the data size of the content increases with the bitrate level at the same time. The size of the tile z of version d is represented as $s_{z,d}(s_{z,D}...>s_{z,d+1}>s_{z,d}>s_{z,d-1}>...s_{z,1})$. And the tiles within the FoV is denoted as the set $\mathcal{Z}_{ug} \subseteq \mathcal{Z}_g$. Thus, the number of tiles within the FoV is denoted as $k = |\mathcal{Z}_{ug}|, u \in \mathcal{U}$.

TABLE I
MATHEMATICAL NOTATIONS

Symbol	Description
$\mathcal{B}, \mathcal{U}$	The set of edge base station and viewers
$\mathcal{G}, \mathcal{D}$	The set of GOP and various bitrate of the VR video
$\mathcal{Z}_g, \mathcal{Z}_{ug}$	The set of tiles and the FoV of viewer u
$\mathcal{M}$	The set of multicast group during delivery process
H_b, H_u	The content cached at edge and requested by viewers
$s_{z,d}$	The size of the tile z of version d
P_b, P_{bg}	The transmit power of edge BS b and the g-th content
$d_{b,u}$	The distance between edge BS b and viewer u
g_b	The maximum number of superposed content of b
$R_{a,b}$	The transmission rate between a and b
c	The computational cost for implementing transcoding
k	Tiles number in FoV of viewer
T_{up}, T_{ud}	The latency of preparation and delivery
$\mathbb{I}_{\{\cdot\}}$	The indicator of $\{\cdot\}$. 1 for true and 0 for false
$\mathcal{O}_b, C_b$	The storage space and computational resource
S, A, R	Symbols about state, action and reward
V_b^π, Q_b	Value-function and the state-action function
Ψ, Υ	Different matching for the edge VR delivery

C. Cache Model

The edge BS is enabled with storage and computation functions. On one hand, it stores the content most likely to be requested by viewers nearby and realizes rapid delivery of the content. On the other hand, the BS has the ability to spend computational resources and transcode the video locally, thus converting the content of version d to the content of a lower version d' ($d'<d$) to meet the requirements of more viewers [8]. The storage space of BS b is represented by $\mathcal{O}_b$. The number of cached high bitrate tiles per GoP is set to k, which is considered the size of FoV. Fewer cached high quality tiles require additional tile requests from the remote server, which puts additional pressure on both network and server. A larger k implies a higher cache storage space share allocated for this viewer from the limited total storage space. Therefore, in a desired solution, the k tiles cover the whole size of viewer FoV. This is to improve efficient storage space utilization and encourages the edge BS to cache beneficial tiles without wasting valuable space. The difference is that these tiles can be nonadjacent and form a virtual viewport. The computing resources of BS b are represented by C_b. The computational cost[9] for implementing the transcoding of content from bitrate

[5]A variety of projection methods can be used including equirectangular, cube maps, pyramid maps, equal-area projection, etc. [39].

[6]The GOPs has segments with fixed duration, e.g. 3 seconds.

[7]DASH has extended its standard to virtual reality with the amendment on Spatial Representation Description (SRD) [40].

[8]The adaptive bitrate (ABR) algorithm deployed at the client determines dynamically the content bitrate according to the state of the buffer and network conditions.

[9]Our test show that the transcoding cost is weakly correlated with the span of bitrate level, but closely correlated with the target bitrate level [36].

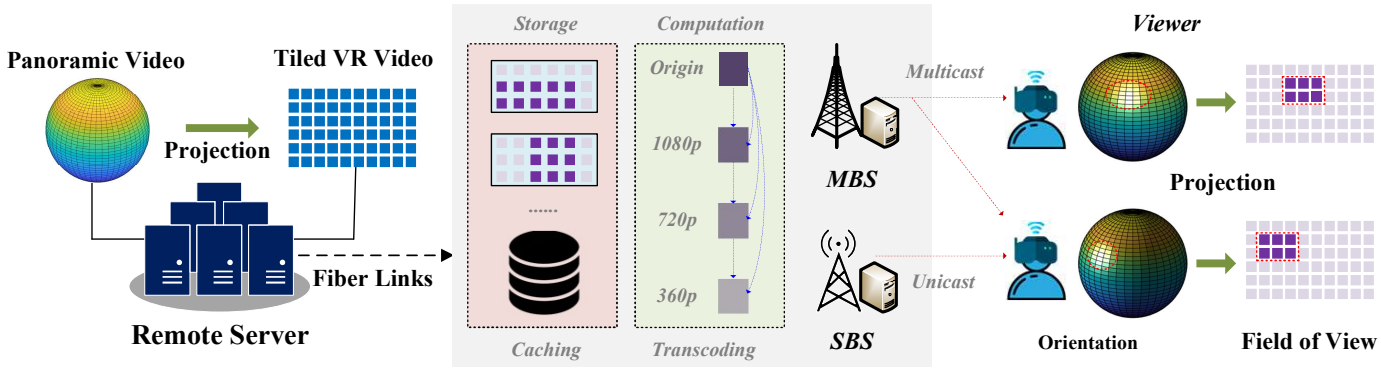


Fig. 2. System Architecture Diagram. The panoramic VR video is processed (projected and tiled) at the remote server and transmitted to the edge network. Edge networks serve as many users as possible through caching and transcoding. The edge BS delivers content to the viewers through multicast scheduling.

d to bitrate d' is denoted as $c_{d,d'}$. This cost can be measured by a transcoding program such as *ffmpeg*[10].

D. Delivery Model

During the delivery process, a NOMA-based edge multicast is considered in this paper. With the help of power domain division multiplexing-based NOMA, edge BSs can transmit a mixture of messages of most g_b contents on the same time-frequency-code resource [50]. The amount of superposed content is considered as the maximum number of concurrency for NOMA-based edge multicast. The maximum transmit power of edge BSs is assumed to be P_b and $\sum_{n=1}^{g_b} P_{bn} = P_b$, where P_{bn} is the transmit power for n-th content.

In relation to the decoding process of NOMA, the data traffic of the VR video service mainly traverses the downlink from edge BS to viewers. Therefore, we mainly discuss the decoding order on the downlink NOMA. We sort the viewer set in terms of the channel gain. If the channel gain of one viewer is greater than that of another viewer, the former viewer is considered to be a stronger signal user and the latter is a weaker signal user. During the decoding process of downlink signals, the viewers with poor channel quality need to be allocated higher transmission power to ensure their basic data rate requirements. Therefore, among viewers sharing a NOMA cluster, the interference to viewers mainly comes from the weaker signal viewers. As such, the decoding order is set as follows.

The weaker signal viewers directly modulate the data (i.e. the signals from strong signal viewers are regarded as interference in this process). The stronger signal viewers first modulate the data of the weak signal viewers, then the successive interference cancelation (SIC) is applied, and finally the data they want to receive is modulated.

Thus, for the viewer u, the received signal-to-interference ratio (SIR) (after successfully decoding the symbols from n-th content) is computed as follows.

$$\mathrm{SIR_u} = \frac{h_b |d_{b,u}|^{-\alpha} P_b}{\sum\limits_{b' \in \mathcal{B} \setminus b} h_{b'} |d_{b',u}|^{-\alpha} P_{b'} + h_b |d_{b,u}|^{-\alpha} \sum\limits_{n'=n+1}^{g_b} P_{bn'}} \tag{1}$$

[10]https://github.com/FFmpeg/FFmpeg

where $d_{b,u}$ is the distance between edge BS b and viewer u. α is the path loss exponent. The links between the viewers and edge BS are assumed to be subjected to Rayleigh fading. h_b is the channel coefficient of the link between edge BS b and the viewer, which follows an exponential distribution with mean one, i.e. $h_b \sim \exp(1)$. Thus, the transmission rate during delivery can be denoted as $R_{b,u} = W_u log_2(1 + \mathrm{SIR_u})$. The content of the data sub-layers can be correctly decoded by viewers through successive interference cancellation (SIC) when the minimum rate limit is satisfied [28], [29].

IV. Problem Formulation

First, in this section, the content service process is summarized as follows. The viewer u sends the content requests $H_u = \{f, d, l\}$ to the edge BS b, where f is the content, d is the version and l is the tiles within FoV. The edge node checks whether the corresponding or higher version of the content is cached. When the content is already cached, it can be quickly and directly delivered to the viewer. If content of higher versions is available, it can also be delivered to the viewer following transcoding. In cases where the content is not cached, the viewer can obtain it from the collaborative BSs through fiber links. Otherwise, the content has to be fetched from the remote server. For avoiding motion sickness, we formalize the VR video service process as a latency minimization problem next. The problem explicitly includes the caching and transcoding latency in the edge network (the delivery preparation phase) and the multicast latency (the delivery execution phase) during the delivery process. The major components of latency considered are communication latency and computational latency.

A. Communication Latency

Communication latency is mainly embodied in two stages. On one hand, the latency appears in fetching data from the remote server if the content is not hit at edge. Since BS b is connected to the remote server through the optical fiber link, the transmission rate on the link is denoted as $R_{r,b}$, then the latency is denoted as $t_{r,b,z,d} = s_{z,d}/R_{r,b}$. Similarly, the rate between BS b and another BS b' is denoted as $R_{b',b}$ and the latency is $t_{b',b,z,d}$.

On the other hand, the latency appears when delivering the content from edge to viewers. In order to reduce the load

at edge network and the average latency of viewers in the region, the delivery process is considered as a combination of unicast and multicast. The viewers who watch the same content are bundled into a multicast group. The set of multicast groups is denoted as $\mathcal{M} = \{1, 2, ..., m, ..., M\}$. $\mathcal{M}_m = \{\mathcal{M}_{m,\mathcal{U}}, \mathcal{M}_{m,\mathcal{Z}}, \mathcal{M}_{m,D}\}$, where $\mathcal{M}_{m,\mathcal{U}}$ is the group member set and $\mathcal{M}_{m,\mathcal{Z}}$ is the tile set of shared vision. $\mathcal{M}_{m,\mathcal{D}}$ is the version requested. And the transmission rate $R_{b,d} = R_{b,u}$ is employed to complete the delivery. The delivery latency is $t_{b,u,z,d} = s_{z,d}/R_{b,d}$.

B. Computational Latency

In the case of caching higher versions of content, the viewer can be served by transcoding, which avoids the communication latency from the remote server. Instead, it takes a certain amount of computing resources and causes the associated computational latency. Consider the content cached by BS b as $H_b = \{f, d', l\}$. The computational resources required for transcoding can be denoted as $c_{b,u,d',d} = c_{d,d'}$. The corresponding computation latency is $t_{b,u,c} = c_{b,u,d',d}/C_b$.

Therefore, the total latency T_u of the service process for viewer u can be summarized as follows, including the latency of preparation phase T_{up} and delivery execution phase T_{ud}:

$$\begin{aligned} T_u =& T_{up} + T_{ud} \\ =& \underbrace{\mathbb{I}_{\{u,e\}} t_{r,b,z,d} + \mathbb{I}_{\{u,b,b'\}} t_{b',b,z,d} + \mathbb{I}_{\{u,b\}} t_{b,u,c}}_{preparation} + \underbrace{t_{b,u,z,d}}_{delivery} \end{aligned} \tag{2}$$

where $\mathbb{I}_{\{u,e\}}$ indicates whether the content requested by u should be delivered by the remote server, $\mathbb{I}_{\{u,b,b'\}}$ means that the content can be provided (directly or by transcoding) by another collaborative BS b', other than BS b, and $\mathbb{I}_{\{u,b\}}$ shows whether the content can be located (directly or by transcoding) on BS b. There exists the following relationship for each indicator:

$$\begin{cases} \mathbb{I}_{\{u,b\}} = \sum\limits_{\forall \mathcal{O}_{bi} \in S_b} \mathbb{I}_{\{\mathcal{O}_{bi}, Q_u\}} & \text{(3a)} \\ \mathbb{I}_{\{u,b,b'\}} = \left(1 - \mathbb{I}_{\{u,b\}}\right) \sum\limits_{\forall b' \in \mathcal{B}, b' \neq b} \mathbb{I}_{\{u,b'\}} & \text{(3b)} \\ \mathbb{I}_{\{u,e\}} =! \left(\mathbb{I}_{\{u,b\}} \&\& \mathbb{I}_{\{u,b,b'\}}\right) & \text{(3c)} \end{cases}$$

where $\mathbb{I}_{\mathcal{O}_{bi}, Q_u}$ means that $\mathcal{O}_{bi}$ is the content requested by u. $\mathbb{I}_{u,b} \in [0,1], \forall b \in \mathcal{B}$ means that only one version of each content can be cached in edge BS. $\&\&$ is the AND operator and is used to combine the two cases described in sub-equations (3a) and (3b). ! is the NOT operator and is used to denote an opposing logical value.

The average time latency of the viewer at edge is then expressed as follows.

$$\begin{aligned} \overline{T}_{\mathcal{U}} &= \frac{1}{U} \sum_{\forall u \in \mathcal{U}} T_u \\ &\overset{(a)}{=} \frac{1}{U} \sum_{\forall u \in \mathcal{U}} T_{up} + \frac{1}{U} \sum_{\forall u \in \mathcal{U}} T_{ud} \end{aligned} \tag{4}$$

Through the equation item (a), the average latency is decomposed into the average preparation latency and the average delivery latency. The transformation can be extended by formula (2). Thus, the average latency minimization problem is formulated.

$$\begin{aligned} \min \ & \overline{T}_{\mathcal{U}} \\ s.t. \ & b \in \mathcal{B}, d \in \mathcal{D}, u \in \mathcal{U} \\ & \mathbb{I}_{u,b} \in \{0,1\}, \mathbb{I}_{u,b,b'} \in \{0,1\}, \mathbb{I}_{u,e} \in \{0,1\} \\ & \sum_{u=1}^{U} \mathbb{I}_{\mathcal{B}_u, b} c_{b,u,d',d} \leq C_b \\ & \sum_{g \in \mathcal{G}} \sum_{z \in \mathcal{Z}_g} \mathbb{I}_{z,b} s_{z,d'} \leq \mathcal{O}_b \end{aligned} \tag{5}$$

where $\mathbb{I}_{\mathcal{B}_u, b}$ is the indicator of $b = \mathcal{B}_u$ and $\mathbb{I}_{z,b}$ denote whether the tile z is cached in BS b.

In the problem (5), we have two undetermined variables. The first variable is where the content can be fetched from, which is a binary variable (i.e. base station or remote server). The second indicates how to deliver the content to viewer, which is a discrete variable (i.e. unicast delivery, multicast delivery from MBS or SBS). In fact, as shown in equation (2) and equation (4), the optimization process is primarily divided into two separate phases, which are associated with two subproblems: delivery preparation phase (i.e. caching and transcoding) and delivery execution phase. Next these two subproblems are focused on and the goal is to determine optimal values for the two variables, respectively.

V. Cache and Transcoding-assisted Delivery Preparation Phase

This section describes the cache and transcoding-assisted delivery preparation phase. The cache of each BS changes following the viewer requests at every moment. In order to model this complex dynamic process, we represent it as a Markov decision process (MDP).

A. Markov Decision Process

The MDP of edge BS is summarized as a 4-tuple $\mathcal{K} = \{S, A, P, R\}$.

- $S = \{S_s, S_r, S_v, S_l\}$ is the state, including the current cache state S_s, the request information from viewer S_r, the tile information for various versions S_v and the capacity limit $S_l = \mathcal{O}_b$. $S_s = \{s_{s1}, s_{s2}, ..., s_{sz}\}, \forall s_{sz} \in (0,1)$ is a vector with the size of tiles in the video (e.g. $G * Z$). Each element is used to represent whether the tile is cached (i.e. 1 for cached and 0 for no hit). $S_r = \{s_{r1}, s_{r2}, ..., s_{rz}\}$ is used to denote the request information. $s_{rz} \in [0, +\infty)$ is denoted the requested times for the tile in the past h slot. $S_v = \{s_{v1}, s_{v2}, ...s_{vD}\}$. Each element is the unit size of a different version.
- $A = \{A_{gz}\}, A_{gz} \in [0, D], \forall g \in \mathcal{G}, \forall z \in \mathcal{Z}$ is the cache action for the video content with the unit of tile. $A_{gz} = 0$ mean that the tile is not cached. Other options mean that the corresponding version of the tile is cached.

- $P\left(s'|s,a\right)$ is the transition probability from state s to another state s' through the action a. In other words, P represents the impact and the model of the environment.
- R is the reward. It is determined by the state and action (i.e. $S \times A \rightarrow R$), including the service latency of the viewer within the range of the edge BS and the capacity penalty Ω. The former is denoted as $\overline{T}_{bp} = \frac{1}{N}\sum_{\forall u,\mathcal{B}_u=b} T_{up}, \forall b \in \mathcal{B}$, where N is the number of viewers within the range of BS b. The latter is a negative constant for punishing the cache overflow phenomenon.

B. Markov Game

In fact, due to the cooperation between BSs, the caching decision of each BS has a significant impact on the service performance of other BSs. In other words, BS b receives rewards not only from its own action, but also from the actions of other BSs (e.g. b'). Therefore, we extend the MDP of a single agent to a Markov Game (MG) for a multi-agent setting by taking the relation between agents into account. A MG is denoted as $\mathcal{V} = \{\mathcal{B}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{R}, \gamma\}$. $\mathcal{B}$ is the set of participant (i.e. edge BSs), each of whom is also referred to as an agent. $\boldsymbol{S} = \{s_b\}, \forall b \in \mathcal{B}, s_b \in S$ is the joint state space. $\boldsymbol{A} = \langle a_1, a_2, ..., a_B \rangle, a_b \in A$ is the joint actions. $\boldsymbol{R} = \{R_b\}, \forall b \in \mathcal{B}, R_b \in R$ is the immediate joint reward. γ is the discount factor. In a MG, the environment faced by each agent is non-stationary, affected not only by the evolution of the underlying communication system, but also by the decisions made by other agents which are improving their strategies. This makes most single agent-based schemes ineffective.

It is worth noting that the benefits of caching are generally not clear in the short term. Actually, the edge BS needs a caching strategy that can obtain the superior long-term cumulative benefits. On the other hand, the reward of each BS is different from each other. For the convenience of expression, the utility of edge is denoted as $R_e = \sum_{b\in\mathcal{B}} R_b$. Further, the team-average reward is defined.

Definition 1. ***Team-Average Reward*** *allows more heterogeneity among agents and facilitates the development of decentralized algorithms. It is defined next.*

$$\overline{R} := B^{-1} \cdot \sum_{b\in\mathcal{B}} R_b \tag{6}$$

Thus, the long-term reward can be denoted as follows.

$$\begin{aligned} \mathcal{R} =& \frac{1}{T}\lim_{T\to\infty}\sum_{t=1}^{T}\overline{R} = \frac{1}{TB}\lim_{T\to\infty}\sum_{t=1}^{T}\sum_{b\in\mathcal{B}} R_b \\ =& \frac{1}{TBN}\lim_{T\to\infty}\sum_{t=1}^{T}\sum_{b\in\mathcal{B}}\sum_{\forall u\in\mathcal{U},\mathcal{B}_u=b}\left(T_{up} + \mathbb{I}_{bo}\Omega\right) \end{aligned} \tag{7}$$

where $\mathbb{I}_{bo}$ is the indicator for overflow.

The purpose of the agent b is to find a best policy $\pi_b : \boldsymbol{S} \rightarrow a_b$ to maximize the benefits. And the joint policy of all agents is denoted as $\boldsymbol{\pi}\left(\mathbf{a}|\mathbf{s}\right) = \prod_{b\in\mathcal{B}} \pi_b\left(a_b|\mathbf{s}\right)$, where $\mathbf{s} \in \boldsymbol{S}, \mathbf{a} \in \boldsymbol{A}$. For the sake of representation, let $-b$ denote the set of all agent in $\mathcal{B}$ except b next. For example, A_{-b} represent the action of BS other than b. π_{-b} is the policy set except b. Hence, the value-function of agent i is defined as follows.

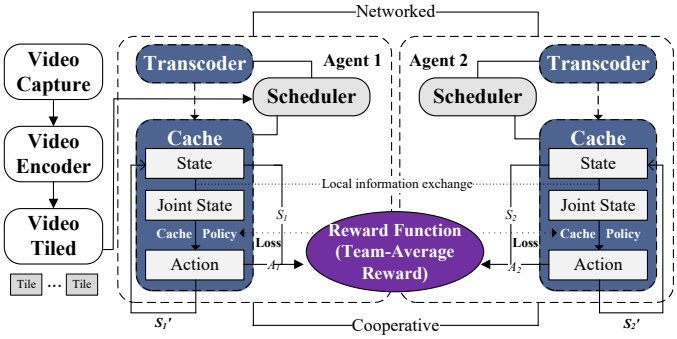


Fig. 3. Cooperative cache. Each agent can make the cache decisions according to the joint state.

$$V_b^{\boldsymbol{\pi}}\left(s\right) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \mathcal{R}_t\left(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}\right) | a_t^b \sim \pi_b\left(\cdot|\boldsymbol{s}_t\right), \forall a_t^b \in \boldsymbol{a}_t\right] \tag{8}$$

where $\boldsymbol{s}_t$ and $\boldsymbol{a}_t$ are the instances of $\mathbf{s}$ and $\mathbf{a}$ at the t-th slot, respectively.

As can be seen from the above, in order to obtain the maximum benefits, b not only needs to consider the service status of the viewers within its own scope, but also needs to consider the impact of its decision on the viewers in the range of other BSs. This makes the evolution even more dynamic.

In this case, the purpose of an agent is to find an optimal strategy π_b^* to obtain the maximum value. However, the state is ever-changing and the relationship between agents is coupled, which makes very challenging for an equilibrium to be achieved. On one hand, the multi-dimensional and multi-scale characteristics of the state space makes the solution space of the problem very large. The problem is difficult to be solved within polynomial time by an optimal algorithm. On the other hand, the complex time-varying environment makes the heuristic algorithm have to repeat the solution process, which reduces the flexibility. Next, we introduce a multi-agent reinforcement learning method to handle this problem and try to find the equilibrium point.

C. Multi-Agent Deep Reinforcement Learning

In order to solve the non-stationary nature of the environment, in this subsection, we propose a ***N**etworked **D**ecentralized paradigm of **M**ulti-agent **A**ctor **C**ritic* algorithm (***NDMAC***), based on the framework of centralized training and decentralized execution [43]. In the networked decentralized multi-agent reinforcement learning, multiple agents perform decisions in a common environment without the coordination of any central controller. This also allows each agent to exchange information with their neighbors over the available communication networks, as shown in Fig. 3.

1) Actor-Critic architecture: Actor-Critic combines the advantages of policy-based and gradient-based methods, which can realize simultaneously the off-policy learning and the

Algorithm 1 Multi-agent Actor-Critic Algorithm

Input: The edge BSs $\mathcal{B}$. The viewers $\mathcal{U}$. The information of the video $\mathcal{G}, \mathcal{D}, \mathcal{Z}$. The information of the BSs C, $\mathcal{O}$.
Initialization: For each agent $b \in \mathcal{B}$, the initial neural network φ, ϑ is set with parameter θ_b, μ_b, respectively. The state $\boldsymbol{s}$ and joint actions $\boldsymbol{a} = \{a_1, ..., a_B\}$ are set.
$t = 0$;
while $t \leq$ max-epidode-length **do**
  / Action */*
  for Agent $b \in \mathcal{B}$ **do**
    Collect the state information $\boldsymbol{s}$ from the edge BSs and generate the action $a_b = \varphi_b(\boldsymbol{s})$.
    Execute the cache action a_b for regional viewer in current slot and generate the new state $\boldsymbol{s}'$. And the transition is stored in replay buffer.
    Obtain the local reward R_b and share it with the other edge BSs.
  end for
  The team-average reward is obtained based on equation (7).
  / Evaluation and Update */*
  for Agent $b \in \mathcal{B}$ **do**
    Critic: Calculate the target Q value and the loss based on equation (10)-(11).
    Actor: The gradient of the expected return of actor φ_b is obtained.
  end for
end while

rapid update. Specifically, *Actor* $\boldsymbol{\varphi}$ is used to output the action and *Critic* $\boldsymbol{\vartheta}$ is used to judge the advantages of the action. And θ and μ are the parameter of these two networks, respectively. The above two structures form an agent together. Thus, *NDMAC* includes B *Actor* and *Critic*, respectively. And for any critic ϑ_μ^b, in order to measure how good the action is in current state, the action-value Q_b, which is also called Q value, is given as follows.

$$Q_b\left(\boldsymbol{s}, \boldsymbol{a} | \vartheta_\mu^b\right) = \mathbb{E}\left[\mathcal{R}_t + \gamma Q_b\left(\boldsymbol{s}', \boldsymbol{a}'\right)\right] \tag{9}$$

where $\boldsymbol{s} = \{s_b^t, s_{-b}^t\}$ and $\boldsymbol{a} = \{a_b^t, a_{-b}^t\}, \forall a_b^t = \varphi_\theta^b(\boldsymbol{s})$. Similarly, $\boldsymbol{s}'$ and $\boldsymbol{a}'$ denote the corresponding vector in $t+1$ slot.

As shown above, the Q value is a recursive expression. In this paper, the neural network (*Critic*) is used as a fitting function to obtain the Q value. The loss during the back propagation is:

$$L^{(t)}\left(\vartheta | \mu, \mu'\right) = \mathbb{E}\left[\left(y(t) - Q_b(\boldsymbol{s}, \boldsymbol{a}) | \vartheta_\mu\right)\right]^2 \tag{10}$$

where $y(t)$ is the target Q-value and expressed as follows.

$$y(t) = \mathbb{E}[(1-\beta) Q_b(\boldsymbol{s}, \boldsymbol{a} | \vartheta_\mu) + \beta\left(R_t + \gamma Q_b(\boldsymbol{s}', \boldsymbol{a}' | \vartheta_\mu')\right)], \tag{11}$$

In addition, we introduce the performance objective J to measure the advantage of the actor policy, as follows.

$$\begin{aligned} J_\delta\left(\varphi_\theta\right) &= \int_S \rho^\delta(\boldsymbol{s}) Q(\boldsymbol{s}, \boldsymbol{a}) ds \\ &= \mathbb{E}_{\boldsymbol{s}\sim\rho^\delta}\left[Q\left(\boldsymbol{s}, \varphi_\theta^1(\boldsymbol{s}), \varphi_\theta^2(\boldsymbol{s}), ..., \varphi_\theta^B(\boldsymbol{s})\right)\right] \end{aligned} \tag{12}$$

where ρ^δ is the PDF of state. And $\varphi_\theta^b, \forall b \in \mathcal{B}$ is the parameterized neural network. Then, the gradient of the expected return of agent b is denoted as follows.

$$\begin{aligned} \nabla_\theta J_\delta\left(\varphi_b\right) &\overset{(1)}{\approx} \mathbb{E}_{\boldsymbol{s}\sim\rho^\delta}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\varphi}(\boldsymbol{a}|\boldsymbol{s}) \cdot \nabla_{\boldsymbol{a}} Q(\boldsymbol{s}, \boldsymbol{a} | \boldsymbol{\mu})\right] \\ &\overset{(2)}{\approx} \frac{1}{\chi} \sum_b^\chi \left[\nabla_{\theta_b} \varphi_b(a_b|\boldsymbol{s}) \cdot \nabla_{a_b} Q(\boldsymbol{s}, a_1, ..., a_B | \mu_b)\,|_{a_b=\varphi_b(\boldsymbol{s})}\right] \end{aligned} \tag{13}$$

where χ is the number of sampled minibatch from the replay buffer. As shown in equations (10)-(14), the decision and update of policy of any agent needs to know the actions and state of other agents, which is not a particularly restrictive assumption [44]. On the other hand, the edge BS within the area can realize fast communication and information exchange through an optical fiber link. Similar to [45], [46], the signaling overhead is ignored here. The detailed algorithm is introduced in ***Algorithm 1***.

This solves the pre-delivery optimization problem. Next, we consider the delay optimization in the edge delivery process.

VI. Multicast and Edge-supported Delivery Execution Phase

This section focuses on the delivery execution phase in which the cached or transcoded tiled video is delivered to the viewer. In an environment with large-scale user requests, the delivery process puts great communication pressure on the edge network, and may affect the QoE of viewers. As discussed earlier, the shared FoV of viewers provides an opportunity for multicast usage. However, on one hand, the viewers requesting the same tiles are scattered in the range of various SBSs, which increases the difficulty of multicast scheduling. On the other hand, the multicast ability of the SBS within a time slot is also limited. Additionally, the selection of multicast content is also a key factor affecting user QoE. This is actually a combinatorial problem between base station and multicast group. In this context, a two-tier multicast scheduling scheme based on matching theory [47], [48] is proposed next for solving the scheduling problem mentioned above.

A. Matching Formulation for Multicast Delivery

Matching theory is a powerful economic tool which can describe the mutually beneficial relationships between participants [49]. In the context of this work, it can help find a tractable solution to improve the delivery between edge BSs and multicast groups. The designed delivery matching is given by a 4-tuple $\{\mathcal{B}, \mathcal{M}, \succ_b, \succ_m\}$, where $\succ_b$ and $\succ_m$ are the preference profile of edge BS and multicast group, respectively. $\mathcal{M} = \{1, 2, ..., m, ...\}$ is the set of multicast groups. The match is defined as follows.

Definition 2. *Match Υ is the solution for associating edge BS and multicast group, which can be defined as the mapping*

function $\mathcal{B} \cup \mathcal{M} \rightarrow 2^{\mathcal{B} \cup \mathcal{M}}$*, where* $2^{\mathcal{B} \cup \mathcal{M}} = \{e | e \subseteq \mathcal{B} \cup \mathcal{M}\}$ *is the set of all the subsets of* $\mathcal{B} \cup \mathcal{M}$*.*

- $|\Upsilon(m)| \leq g_b$ *and* $\Upsilon(m) \subseteq \mathcal{B} \cup \phi$*,* $m \in \mathcal{M}$
- $|\Upsilon(b)| \leq 1$ *and* $\Upsilon(b) \in \mathcal{M} \cup \phi$*,* $b \in \mathcal{B}$
- $\Upsilon(b) = m$ *if and only if* $\Upsilon(m) = b$

where $2^{(\cdot)}$ is the set of all subsets. $|\Upsilon(\cdot)|$ is the cardinality of the match result. $\Upsilon(m) = \phi$ means that the content corresponding to multicast group m should be delivered through unicast by its bound BS. $\Upsilon(b) = \phi$ means that edge BS b does not perform multicast in current slot.

B. Preference Profile for Participant

The preference profile is a complete, reflexive and transitive binary relation between two rational participants sets $\mathcal{B}$ and $\mathcal{M}$. It can be used to characterize the selection tendency of unilateral participants for matching. The preference relationship $\succ_i, i \in (b, m)$ is used to describe the priority of participants during the matching process. For the participants x, y on one side (e.g. edge BSs or multicast groups) and the participant i on the other side, the relation $x \succ_i y$ is defined next.

$$x \succ_i y \Leftrightarrow U_i(x) > U_i(y) \tag{14}$$

where $U_i(\cdot)$ is the utility function. The preference profile is then introduced as follows.

1) Preference Profile for VR Multicast Group: Due to the selfishness of the participants, the multicast group expects the matching edge BS to minimize the delivery delay of the multicast group, i.e. maximizing the transmission rate for the member within the multicast group ($\min T_{ud} = \min \frac{s_{z,d}}{R_{b,u}} \rightarrow \max R_{b,u}$). Meanwhile, the multicast group can only bind one edge BS at the same time. Thus, the utility function of a multicast group can be expressed as: $U_m(b) = \max_{b \in \mathcal{B}} \frac{1}{|\mathcal{M}_m|} \sum_{u \in \mathcal{M}_m} R_{b,u}$. The multicast group sorts the edge BSs according to the utility and selects the one that could provide the maximum average transmission rate as the best match.

2) Preference Profile for Edge BS: Similarly, for the purpose of reducing the average latency in edges, the edge BS sorts all the multicast groups and selects the candidate multicast group according to its own multicast capacity. In this context, an utility function of BS is defined as a function of transmission rate similar to the preference profile for multicast group, i.e. $U_b(m) = \max_{m \in \mathcal{M}} avg\{\sum_{u \in \mathcal{M}_m, b = \mathcal{B}_u} R_{b,u}\}$.

It is worth noting that a multicast group includes the viewers from various SBSs and is just globally put together by MBS. On the other hand, due to the limited range of SBS, in generally a SBS cannot meet the service requirement for all members in a multicast group. Therefore, the matching theory cannot be applied directly to the current situation. Instead we propose the following two-tier matching algorithm as a solution.

C. A Two-Tier BS-Multicast Group Matching Algorithm

The idea behind the proposed two-tier BS-Multicast Group Matching Algorithm includes the following two aspects. First, according to the multicast capacity of BS, we split the edge into two tiers (MBS multicast and SBS multicast) to implement sequential matching. Next, due to the various sources of the members of the multicast group $\mathcal{M}$, it is further divided into several subgroups. The set of the subgroups is denoted as $\mathcal{M}_d$.

Algorithm 2 Two-Tier Matching Algorithm

1: ***Input:*** The edge BSs $\mathcal{B}$. The multicast group $\mathcal{M}$.
2: */* Preference Profile Establishment */*
3: ***Initialization:*** For each edge BS $b \in \mathcal{B}$ and each multicast group m, the preference profile L is established according to its utility function.
4: */* MBS Matching Stage */*
5: MBS traverse its preference profile to select the global multicasting content m'.
6: Multicast group delete MBS in their preference profile.
7: Multicast subgroup is generated according to the binding relationship.
8: **while** $\forall b \in \mathcal{B}, |\Upsilon(b)| = 1$ or $\forall m \in \mathcal{M}, |\Upsilon(m)| = 1$ **do**
9: */* SBS Matching Stage */*
10: **for** BS $b \in \mathcal{B} \backslash \{1\}$, multicast group $m \in \mathcal{M} \backslash \{m'\}$ **do**
11: b and m select the multicasting content with the highest priority, respectively.
12: **if** $\Upsilon_b = m$ and $\Upsilon_m = b$ **then**
13: The matching between b and m is successful.
14: **end if**
15: **end for**
16: **end while**

The proposed BS-Multicast Group Matching algorithm, based on the Gale-Shapley deferred acceptance (DA) algorithm, is detailed in ***Algorithm 2***. In first tier (MBS multicast), according to the utility function and multicast capability, MBS ranks the set of multicast groups according to the lowest latency. Similarly, the multicast group sorts the list of BSs based on the utility function and outputs request preferences. If both sides are mutually selected, the match is considered successful. It is worth mentioning that no SBS participates in the match until MBS completes the match. It can be seen that this process is essentially a seller market of MBS, which can be simplified as an unilateral matching of MBS in the first stage. Next, in the second tier (SBS multicast), the multicast groups are divided into subgroups and bilateral matching is performed. Both the BS and the multicast group select the corresponding results according to preference profiles as shown in line 9-15. And in line 8, it determines whether to enter the next round according to the matching results.

D. Stability and Pareto Optimality

In this subsection, the performance of the two-tier matching is analyzed, including stability and optimality. First, the definition of stability is given.

Definition 3. ***Stability.*** *A match has entered the state of stability if and only if there are no blocked pairs* $(b, m) \in \mathcal{B} \times \mathcal{M}$

such that

$$(b \succ_m \Upsilon_m) \wedge (\exists m' \in \Upsilon_b, m \succ_b m') \tag{15}$$

Proposition 1. *The two-tier delivery matching can converged to the stability.*

Proof: To prove the proposition, we adopt the contradiction. Consider the matching produce a result Υ with a blocking pair (b, m), it means that equation (15) is established. For $b \succ_m \Upsilon_m$, the multicast group m will adjust its selection to b rather than the current matching result Υ_m according to its preference profile. At the same time, $m \notin \Upsilon_b$ means that $\forall m' \in \Upsilon_b, m' \succ_b m$, which violates the right item of equation (15). Thus, the proposition is proved. ∎

Definition 4. ***Pareto-Optimal.*** *The matching result Υ is Pareto optimal if and only if there does not exist other matching Ψ satisfying equation (16a) and one of the conditions from equations (16b) and (16c).*

$$\Psi_m \succsim_m \Upsilon_m, \forall m \in \mathcal{M}; \Psi_b \succsim_b \Upsilon_b, \forall b \in \mathcal{B}. \tag{16a}$$

$$\exists b \in \mathcal{B}, \Psi_b \succ_b \Upsilon_b \tag{16b}$$

$$\exists m \in \mathcal{M}, \Psi_m \succ_m \Upsilon_m \tag{16c}$$

where equation (16a) means that the matching is better than or equal to the original matching Υ for all participants. Equations (16b) and (16c) guarantee that there exists a participant that can earn more benefit from the matching Ψ. The latter is also called the Pareto improvement.

It is worth noting that Pareto optimality is a relative concept. The comparison relation is established on a partial ordered set, and participants can improve their own benefits on the basis of not harming the benefits of others. Affected by the rational setting, the participants never take actions that harm their own interests, even if the actions can improve the total return. This is different from the social welfare maximization.

Proposition 2. *The two-tier delivery matching is Pareto-optimal.*

Proof: Let us assume the stable matching Υ and a new matching Ψ exists. The latter is organized by the selection adjustment from the multicast group m and m' (i.e. $\Psi = \Upsilon - (m, \Upsilon_m) - (m', \Upsilon_{m'}) + (m, \Upsilon_{m'}) + (m', \Upsilon_m)$). For convenience, the utility of the matching can be simply denoted as $U(\cdot) = \sum_{m \in \mathcal{M}} \sum_{b \in \mathcal{B}} \mathbb{I}_{m,b} U_m(b)$, where $\mathbb{I}_{m,b}$ is the indicator for $\Upsilon_b = m$. According to the concept of Pareto improvement, obviously, $U(\Psi) > U(\Upsilon)$. From the definition of preference profile, the higher ranked element is selected by the algorithm. Due to the fact that $\Psi_m \succ_m \Upsilon_m$, Ψ_m must be contacted first for m and rejected. Similarly, $\Psi_{m'}$ is contacted earlier than $\Upsilon_{m'}$ for m' and rejected as well. It can be inferred that $U_{\Psi_m} < U_{\Upsilon_m}$ and $U_{\Psi_{m'}} < U_{\Upsilon_{m'}}$, which violate the definition of Pareto improvement. Thus, the Pareto-optimal of the matching can be proved. ∎

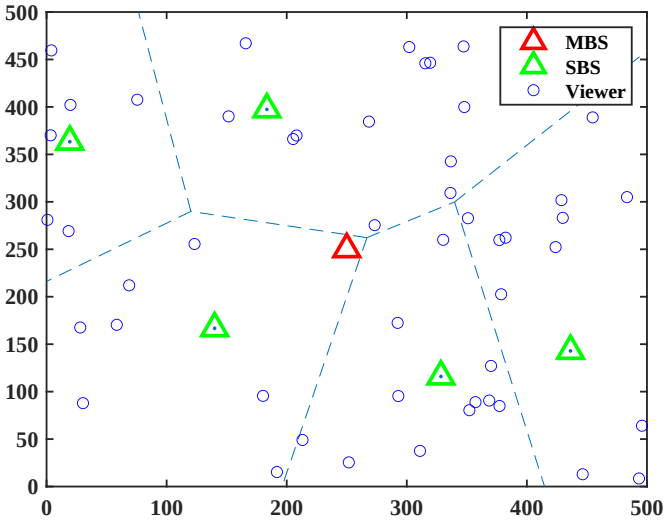


Fig. 4. Experimental Environment. The location of viewers and SBSs obey Poisson distribution. MBS is set in the center of the scene.

E. Completeness and Complexity Analysis

In order to demonstrate the practicability of the scheme, the completeness is demonstrated and the complexity of the algorithm is quantified in this section. According to [51], the two-tier algorithm always finds a solution (i.e. Pareto optimal here) if a solution exists or otherwise it determines that a solution does not exist within a finite amount of time. Thus, the completeness of the algorithm is obtained. First, the multicast group information and the edge BS information are collected by MBS and sorted by preference profile. Thus, according to the standard sorting algorithm, the sorting complexity of multicast group is $O(BlogB)$, and the sorting complexity of edge BS is $O(MlogM)$. In addition, the two-tier approach is applied in the matching phase and the worst-case performance is considered. On one hand, due to the advantages of coverage during multicast, MBS can select multicast groups with the highest priority according to the preference profile. On the other hand, multicast subgroup generation relies on the number of viewers with the complexity of $O(U)$. Finally, during the matching process, the continuous conflicts between BSs occur in the worst case and the complexity is $O(B)$. Therefore, the matching algorithm is mainly deployed in MBS, the complexity is $O\left(B^2 logB + U\right)$. SBSs mainly complete preference sorting, and the complexity is $O(MlogM)$.

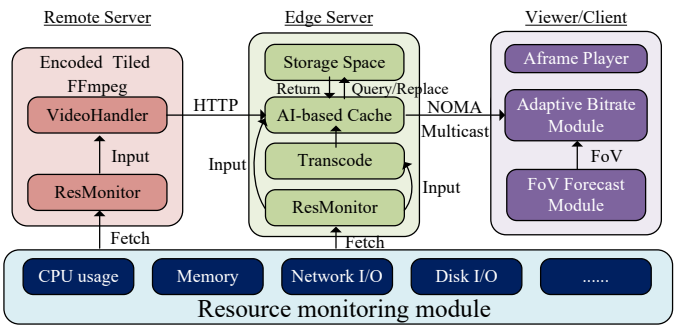


Fig. 5. Testbed architecture which includes three components: Remote Server, Edge Server and Client Viewer.

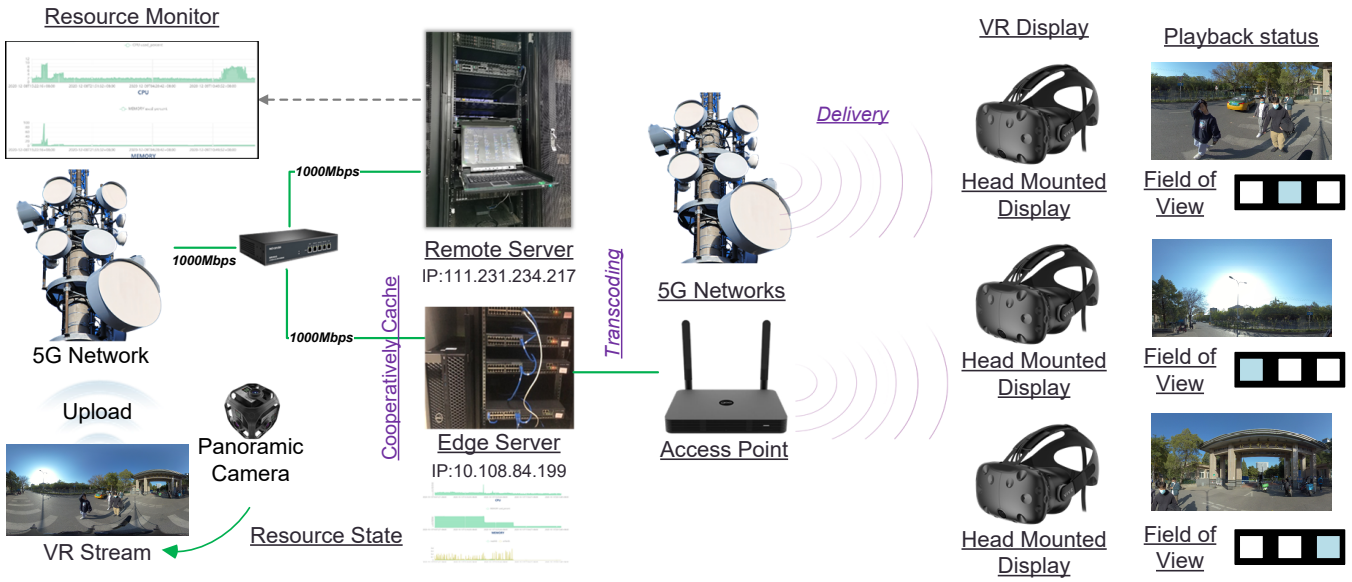


Fig. 6. The VR video recorder by a panoramic camera is uploaded to the Remote Server and further transmitted to the Edge Server, which provides VR video services to Client Viewers via cooperative caching and transcoding. Client Viewers transfer playback information to the Edge Server.

VII. Performance Evaluation

In this section, the performance of our proposed solution is evaluated. First, the experimental environment and parameter settings are described. Then, the performance of our solution is compared with several benchmark algorithms.

A. Experimental Environment

The testing environment considered is a two-layer heterogeneous network with an area of $500*500m^2$. The edge SBSs and the viewer are located in the area with the independently homogeneous PPP with a density of 5 and 50, while the MBS is considered placed at the center of the scene. MBS and SBSs are connected to each other with a gigabit bandwidth. Regarding the content, a VR video library is set to contain 30 files. The popularity of the videos is considered to follow a Zipf distribution with a parameter of 2. The length of the VR video clips is set to $60s$. The resolution of the VR video frame is considered 7680*3840 according to the detailed specs of Insta360 Pro2[11]. Each video is further divided into 30 GoPs (i.e. each GoP is set to 2s and contains 60 frames with 30fps). A GoP is split into 4*4=16 tiles. Each tile, with a resolution of 1920*960 is further encoded with the help of the *ffmpeg* library at four different bit rates: 15Mbps, 30Mbps, 60Mbps, and 120Mbps. For a GoP, the transcoding overhead across one level is considered to be 20GHz. For edge BSs, we consider that a cluster of servers with [6,10]GB of storage space and [50,100]GHz of computational resource is deployed in each node. In addition, the arrival rate of the request of viewers follows the Poisson distribution and the request is delivered to the nearest edge BS for processing. According to the available research, in general, only 12-20% of the frame area is visible [5]. For instance, the FoV of HTC Vive[12] is 110 degrees, less than a third of the horizontal field and less than half of the vertical field (i.e. the areas behind, under viewers feet and above viewers head cannot been watched). Thus, in this paper, the scope of the content requested by the viewers is fixed to 20% of the entire frame. The preference for tiles follows a Gaussian distribution from the center to the edge of frame [52], [53]. The experiment was deployed on an *Aframe Player*-based[13] testbed, as shown in Fig. 5. The architecture includes three components: Remote Server[14], Edge Server and Client Viewer. At the remote server, a resource monitoring (*ResMonitor*[15]) module is deployed to obtain the current resource status (e.g. CPU usage, memory, etc.) of nodes (e.g. remote servers or edge servers). The *VideoHandler* module employs FFmpeg to encode, tile and segment VR videos. The content is transmitted from the remote server to the edge server via the HTTP protocol. The Edge Server provides *Transcoding* and *Caching* functions. Transcoding supports caching, where the cache module queries the storage space on request and performs content replacement according to the cache decision. The Edge Server multicasts content to the Client Viewers using NOMA. The client deploys the *FoV Forecast Module* and *Adaptive Bitrate Module*, which provide information about FoV and the bitrate. The client plays the content using the *Aframe Player*.

B. Performance Result Analysis

1) Transcoding-enabled Edge Cache: To test the performance of NDMAC, it was compared with three benchmark solutions: Central Cloud Services (CCS), Random Cache (RC) and Greedy Cache (GC). The content requested by viewers is obtained directly from the cloud in CCS and there is no caching or transcoding strategy at the edge. The edge BS caches the tiles randomly in RC, whereas in GC, the edge BS caches content according to the history information within

[11]https://www.insta360.com/product/insta360-pro2/

[12]https://business.vive.com/cn/product/vive-pro-eye/

[13]The aframe is a opensource VR player, which is opened at https://github.com/aframevr/aframe

[14]The servers are located at Beijing University of Posts and Telecommunications (BUPT).

[15]The *ResMonitor* is also applied in our previous work in [24].

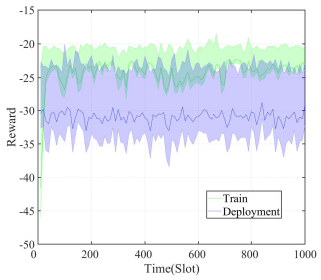


Fig. 7. Average Reward

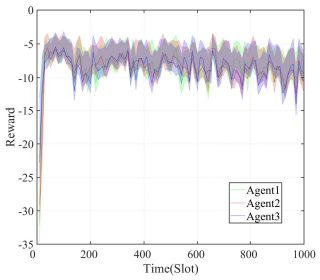


Fig. 8. Reward of Agent

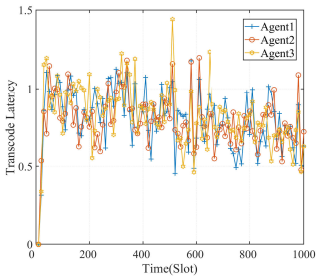


Fig. 9. Transcode Latency of Agent

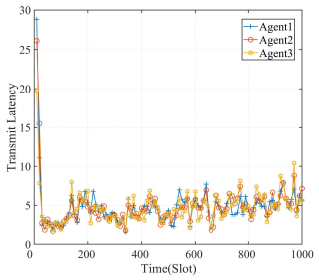


Fig. 10. Transmit Latency of Agent

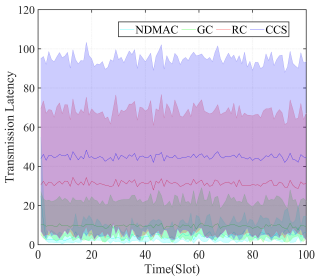


Fig. 11. Transmit Latency

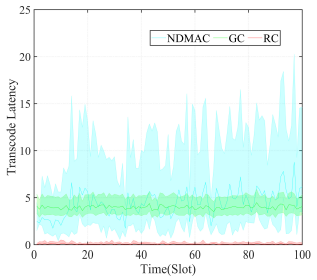


Fig. 12. Transcode Latency

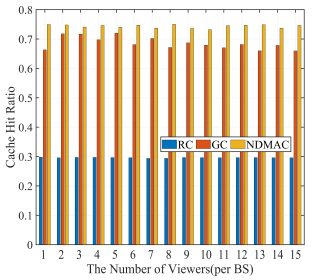


Fig. 13. Cache Hit Ratio

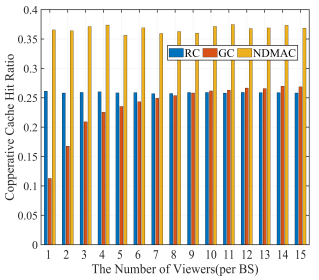


Fig. 14. Cooperative Cache Hit Ratio

8 slots. Before the specific experimental results are analyzed, the concepts of *cache hit ratio* and *cooperative cache hit ratio* are introduced. First, the *cache hit ratio* is the probability of resulting a hit when requesting the content at the edge network, including the associated BS b and other BSs. The *cooperative cache hit ratio* represents the probability that a hit will be recorded at an edge other than the associated BS b when the content is requested. Thus, the *cache hit ratio* is equal to the *cooperative cache hit ratio* plus the *cache hit ratio* at BS b.

First, as shown in Fig. 7, the average team reward of the scheme is plotted. Each point in the solid line is the average of 10 episodes. The extended region is composed of the range difference between the maximum and minimum values within 10 episodes. It includes two processes, training and deployment. Both of them present a relatively stable fluctuating situation. Note that the rewards during training are higher than those in the deployment phase. This is because in the latter case, exploration strategies and critic are removed, so that actors have to deal with complex state spaces on their own. On the other hand, the reward fluctuates more in the latter, which is the result of models responding to complex environments. The change of reward for each agent (edge BS) is shown in Fig. 8. Similar to the situation in the previous figure, the points in the solid line show the average reward of 10 episodes, while the dashed areas show the range of fluctuations. All the agents can enter the equilibrium state, and the reward increases and decreases in a staggered manner with the change of the environment. To find out where the change in reward came from, we conducted the following experiments.

Fig. 9 and Fig. 10 record the change of transcoding latency and transmission latency. The transcoding latency comes from the change of the requests of viewers, which is positively correlated with the difference of bitrate. The transmit latency comes from the overhead of collaboration for edge hits and cloud acquisition. The transcoding overhead of the agent presents an upward trend, while the transmission delay presents a downward trend. This indicates that more requests from viewers are satisfied by the edge network, thus reducing the frequency of the acquisition of content from the cloud. Transmission latency are difficult to eliminate because some content still needs to be retrieved from the cloud, and the collaborative process of the edge BSs always exists.

In order to verify the performance of NDMAC, we further compare it with other schemes. As shown in Fig. 11 and Fig. 12, the transmission latency of each scheme is plotted. Among them, the latency of the situation without cache is the highest of all. Relatively, the random strategy can earn some benefits from the cache mechanism and the greedy cache and NDMAC can further reduce the transmission cost. Regarding transcoding delay, the transcoding delay of the random cache approach is low, due to the low hit rate. The transcoding latency of the greedy strategy is relatively stable, while that of NDMAC is relatively fluctuating.

Finally, the cache hit ratio and cooperative cache hit ratio are illustrated in Fig. 13 and Fig. 14, respectively. NDMAC can achieve a certain degree of benefit. In terms of cache hit ratio, random cache has a low hit ratio. The cooperative caching hit ratio of the greedy strategy shows a certain upward trend with the increase in the number of viewers in the early stage, and tends to be stable in the later stage, indicating that the greedy strategy depends on the amount of observation data. In contrast, NDMAC shows better cooperative characteristics.

2) NOMA-based Multicast Delivery: The shared tiles in the viewport of viewer are considered as paramount for the formation of a multicast group. The multicast capability of each BS is considered to be according to a uniform distribution of [3,6]. The transmitting power is set to 10W. The proposed NOMA-based solution is compared to a FDMA-based multicast scheme and a unicast scheme. The former can only multicast once within one slot, while the latter

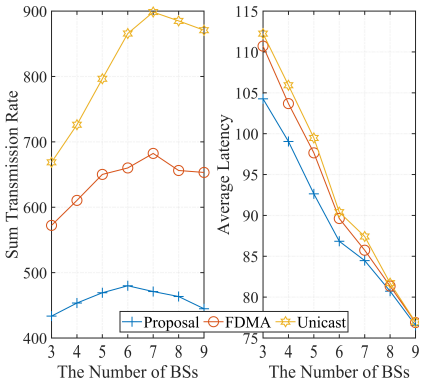


Fig. 15. Performance versus BS Number

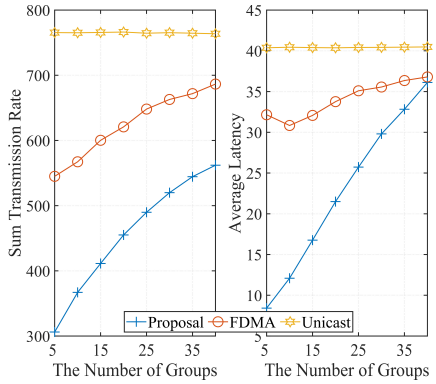


Fig. 16. Performance versus Group Number

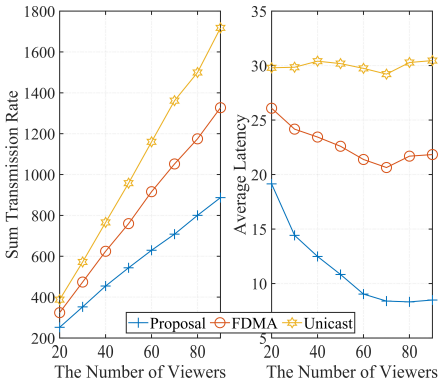


Fig. 17. Performance versus Viewer Number

provides services to viewers with unicast. At the same time, in order to improve the generalization of the proposal and the representativeness of the results, we repeated the algorithm 10 times and took its average as the experimental data. The specific tests are introduced as follows.

As shown in Fig. 15, we first tested the total transmission rate $R_{total} = \sum_{b=1}^{B} \sum_{u=1}^{U} R_{bu}$ of all viewers at the edge during delivery, and observed the change of the total transmission rate over the number of edge BSs, multicast groups, and viewers. First of all, limited by the minimum transmission rate of a member in a multicast group, the total transmission rate of the proposed scheme is the lowest. Unicast-based scheme satisfy individual maximum rates of viewers and therefore its sum of all rates is the highest. As the number of edge BSs increases, the sum of all transmission rates shows an overall upward trend for all schemes. Further it tends to be stable when the number of edge BSs is larger. This is because with the increase of the number of deployed BSs, the load of a single BS is reduced and the transmission rate of serving viewers is improved. However, due to the frequency utilization efficiency of the NOMA-based multicast and the limitation of the minimum transmission rate, the proposed scheme remains in a stable state from beginning to the end. At the same time, the average latency of our proposal is the lowest, as shown at the right of Fig. 15. When the number of BSs is in the range of [3,5], the proposed scheme achieves significant advantages in terms of average latency. As the number increases, the service capacity at edge rises and becomes redundant. The other schemes can benefit from the trend and eventually their latency approaches that of our proposal.

Then, the effect of the number of multicast groups on performance is tested. On one hand, as expected, unicast is not affected by the number of multicast groups. The overall transmission rate is higher in the unicast-based scheme while the average latency is also relatively high. On the other hand, the multicast capacity of FDMA-based scheme is limited by frequency and its performance is inferior to that of the proposed scheme. In fact, with the increase in the number of multicast groups, the members within more multicast group cannot be served via multicast and content has to be delivered by unicast, thus resulting in an increase of the overall transmission rate and latency. Due to the advantages of NOMA in multicast, its performance degrades slower than that of the FDMA-based scheme when the number of multicast groups increases.

In addition, the impact of viewers on performance is also taken into account. As the number of viewers increases, the overall transmission rate increases, and the average latency of viewers shows a different trend. The latency of unicast is relatively stable and the scheme with multicast can get a significant benefit from it. Noteworthy is that the proposed scheme reduces the average latency by 65% compared to the FDMA scheme. In particular, due to the increase in the number of viewers, the average latency of unicast and FDMA-based schemes shows an upward trend within the range of [70,90], while the proposed scheme is still able to maintain a low latency.

And the pareto-optimal of the two-tier matching algorithm is testing in Fig. 18. As a result, the utility of edge BS $U_b = \sum U_b(m), m \in \mathcal{M}$, which is defined in the preference profile for edge BS, is increasing. The gradient of utility tends to go down. Due to the limited multicast capacity of the BS, it cannot provide services for all multicast groups. However, according to the above trend, the Pareto improvement is monotonous, and our scheme can finally achieve Pareto optimality.

Finally, the multicast probability and multicast coverage ratio are assessed. Multicast probability represents the proportion of the number of multicast groups whose requests are satisfied among the total number of multicast groups. The latter represents the ratio of the number of users served by multicast to the total number of viewers. As shown in Fig. 19, the multicast probability of the proposed scheme is higher than that of the FDMA-based scheme. When the number of multicast groups is small, the probability can reach 70%, which is far higher than the delivery capacity of the alternative solution. This leads to the significant advantage in coverage ratio. On the other hand, the smaller the number of multicast groups, the more concentrated are the viewers. The proposed scheme can cover nearly 80% of the viewers when the number of groups is 5. As the number increases, the coverage ratio decreases, but it is still three times as much as that of the FDMA-based scheme. This demonstrates the superiority of the proposed solution for VR content delivery.

VIII. Conclusions

This article proposes a novel transcoding-enabled VR video caching and delivery framework in edge-enhanced next-

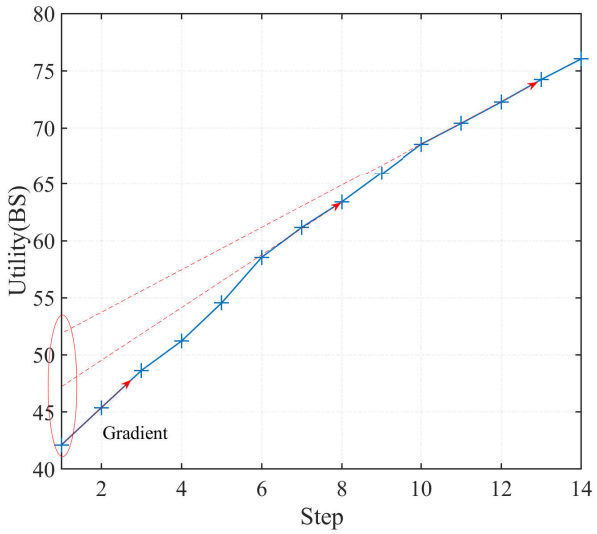


Fig. 18. Pareto optimization of the utility

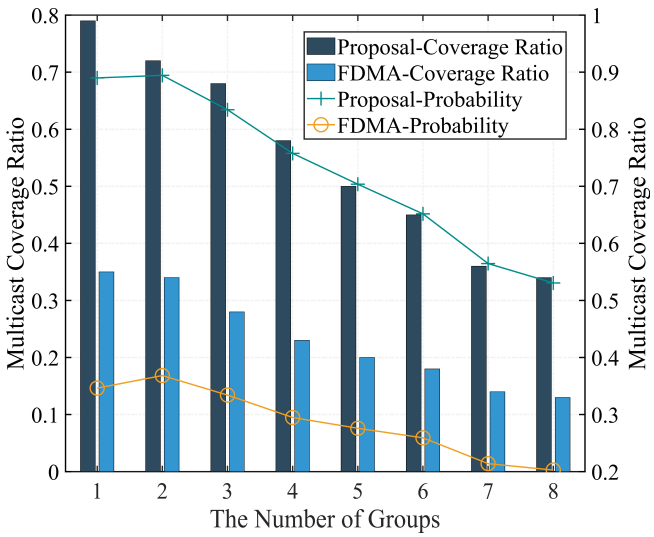


Fig. 19. Multicast Coverage Ratio and Multicast Probability

generation wireless networks. In order to avoid viewer motion sickness and increase their QoE, the VR content delivery is formulated as a latency minimization problem. The problem is split into two sub-phases: caching and transcoding-assisted delivery preparation and edge-supported multicast delivery. First, an edge cooperative caching strategy based on multi-agent deep reinforcement learning is proposed, which makes full use of the storage and computing resources deployed in edge BSs. Then, based on cooperation between edge MBS and SBSs and support from multicast, we design a two-tier BS-multicast group matching algorithm, which provides a practical and feasible solution for VR content delivery. The proposed solution is tested in a series of simulation experiments, and compared with benchmark algorithms. The performance of our proposal outperforms those of state-of-the-art alternative solutions in terms of several metrics, including cache hit ratio, transcoding latency, delivery latency, etc.

REFERENCES

[1] Cisco, Cisco Annual Internet Report (2018−2023) White Paper, 2020. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] M. Zink, R. Sitaraman and K. Nahrstedt, "Scalable 360 Video Stream Delivery: Challenges, Solutions, and Opportunities," in Proceedings of the IEEE, vol. 107, no. 4, pp. 639-650, April 2019.

[3] Shafi, R.; Shuai, W.; Younus, M.U. 360-Degree Video Streaming: A Survey of the State of the Art. Symmetry 2020, 12, 1491.

[4] A. Yaqoob, T. Bi and G. -M. Muntean, "A Survey on Adaptive 360 Video Streaming: Solutions, Challenges and Opportunities," in IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2801-2838, Fourthquarter 2020.

[5] Chao Zhou, Shuoqian Wang, Mengbai Xiao, Sheng Wei, and Yao Liu. 2020. AdaP-360: User-Adaptive Area-of-Focus Projections for Bandwidth-Efficient 360-Degree Video Streaming. In Proceedings of the 28th ACM International Conference on Multimedia (MM '20). Association for Computing Machinery, New York, NY, USA, 37153723.

[6] T. Bi, R. Lyons, G. Fox and G. -M. Muntean, "Improving Student Learning Satisfaction by Using an Innovative DASH-based Multiple Sensorial Media Delivery Solution," in IEEE Transactions on Multimedia.

[7] 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), 2020, pp. 1-6.

[8] G.-M. Muntean, P. Perry, and L. Murphy, A New Adaptive Multimedia Streaming System for All-IP Multi-Service Networks, IEEE Trans. on Broadcasting, vol. 50, no. 1, March 2004, pp. 1-10.

[9] Hongying Liu, Zhubo Ruan, Peng Zhao, Chao Dong, Fanhua Shang, Yuanyuan Liu and Linlin Yang, "Video Super Resolution Based on Deep Learning: A Comprehensive Survey", in CoRR, 2020, arXiv.

[10] P. Xiang, H. Shan, Z. Zhang, L. Yu and T. Q. S. Quek, "NOMA based VR Video Transmissions Exploiting User Behavioral Coherence," 2020 IEEE Wireless Communications and Networking Conference (WCNC), 2020, pp. 1-6.

[11] A. Feriani and E. Hossain, "Single and Multi-Agent Deep Reinforcement Learning for AI-Enabled Wireless Networks: A Tutorial," in IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1226-1252, Secondquarter 2021.

[12] https://github.com/aframevr/aframe

[13] F. Wang, F. Wang, J. Liu, R. Shea and L. Sun, "Intelligent Video Caching at Network Edge: A Multi-Agent Deep Reinforcement Learning Approach," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2020, pp. 2499-2508.

[14] J. Yang, Z. Yao, B. Yang, X. Tan, Z. Wang and Q. Zheng, "Software-Defined Multimedia Streaming System Aided By Variable-Length Interval In-Network Caching," in IEEE Transactions on Multimedia, vol. 21, no. 2, pp. 494-509, Feb. 2019, doi: 10.1109/TMM.2018.2862349.

[15] X. Zhang, T. Lv, Y. Ren, W. Ni, N. C. Beaulieu and Y. J. Guo, "Economical Caching for Scalable Videos in Cache-Enabled Heterogeneous Networks," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 7, pp. 1608-1621, July 2019, doi: 10.1109/JSAC.2019.2916449.

[16] S. Rezvani, N. Mokari, M. R. Javan and E. A. Jorswieck, "Fairness and Transmission-Aware Caching and Delivery Policies in OFDMA-Based HetNets," in IEEE Transactions on Mobile Computing, vol. 19, no. 2, pp. 331-346, 1 Feb. 2020, doi: 10.1109/TMC.2019.2892978.

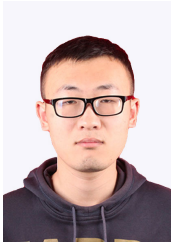
[17] P. Lin, Q. Song and A. Jamalipour, "Multidimensional Cooperative Caching in CoMP-Integrated Ultra-Dense Cellular Networks," in IEEE Transactions on Wireless Communications, vol. 19, no. 3, pp. 1977-1989, March 2020, doi: 10.1109/TWC.2019.2960329.

[18] Georgios Papaioannou and Iordanis Koutsopoulos. 2019. Tile-based Caching Optimization for 360 Videos. In Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '19). Association for Computing Machinery, New York, NY, USA, 171180.

[19] Anahita Mahzari, Afshin Taghavi Nasrabadi, Aliehsan Samiei, and Ravi Prakash. 2018. FoV-Aware Edge Caching for Adaptive 360 Video Streaming. In Proceedings of the 26th ACM international conference on Multimedia (MM '18). Association for Computing Machinery, New York, NY, USA, 173181.

[20] Maniotis P , Thomos N . Viewport-Aware Deep Reinforcement Learning Approach for 360 Video Caching[J]. IEEE Transactions on Multimedia, 2021, PP(99):1-1.

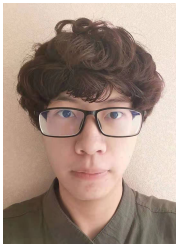
[21] Jacob Chakareski. 2017. VR/AR Immersive Communication: Caching, Edge Computing, and Transmission Trade-Offs. In Proceedings of the Workshop on Virtual Reality and Augmented Reality Network (VR/AR Network'17). Association for Computing Machinery, New York, NY, USA, 3641.
[22] Y. Zhou, F. R. Yu, J. Chen and Y. Kuo, "Video Transcoding, Caching, and Multicast for Heterogeneous Networks Over Wireless Network Virtualization," in IEEE Communications Letters, vol. 22, no. 1, pp. 141-144, Jan. 2018.
[23] X. Chen, Changqiao Xu, M. Wang, Z. Wu, L. Zhong and G.-M. Muntean, "A Universal Transcoding and Transmission Method for Livecast with Networked Multi-Agent Reinforcement Learning," Proceeding of the 2021 IEEE Conference on Computer Communications (INFOCOM 2021), 2021.
[24] X. Chen, Changqiao 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, Early Access Article.
[25] 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," in IEEE Transactions on Communications, vol. 68, no. 4, pp. 2491-2508, April 2020.
[26] 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," in IEEE Transactions on Multimedia.
[27] C. Guo, Y. Cui and Z. Liu, "Optimal Multicast of Tiled 360 VR Video," in IEEE Wireless Communications Letters, vol. 8, no. 1, pp. 145-148, Feb. 2019.
[28] F. Tan, P. Wu, Y. -C. Wu and M. Xia, "Energy-Efficient Non-Orthogonal Multicast and Unicast Transmission of Cell-Free Massive MIMO Systems With SWIPT," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 4, pp. 949-968, April 2021.
[29] Z. Zhang, Z. Ma, Y. Xiao, M. Xiao, G. K. Karagiannidis and P. Fan, "Non-Orthogonal Multiple Access for Cooperative Multicast Millimeter Wave Wireless Networks," in IEEE Journal on Selected Areas in Communications, vol. 35, no. 8, pp. 1794-1808, Aug. 2017.
[30] H. Hao, Changqiao Xu, L. Zhong and D. Wu, "Stochastic Cooperative Optimization for Multicast Scheduling in Heterogeneous and Green 5G Networks," IEEE Transactions on Green Communications and Networking, March 2020.
[31] P. Xiang, H. Shan, Z. Zhang, L. Yu and T. Q. S. Quek, "NOMA based VR Video Transmissions Exploiting User Behavioral Coherence," 2020 IEEE Wireless Communications and Networking Conference (WCNC), 2020, pp. 1-6.
[32] Z. Ding, P. Fan and H. V. Poor, "Impact of Non-Orthogonal Multiple Access on the Offloading of Mobile Edge Computing," in IEEE Transactions on Communications, vol. 67, no. 1, pp. 375-390, Jan. 2019, doi: 10.1109/TCOMM.2018.2870894.
[33] J. Park and M. Bennis, "URLLC-eMBB Slicing to Support VR Multimodal Perceptions over Wireless Cellular Systems," 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1-7, doi: 10.1109/GLOCOM.2018.8647208.
[34] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan and L. Hanzo, "Nonorthogonal Multiple Access for 5G and Beyond," In Proceedings of the IEEE, vol. 105, no. 12, pp. 2347-2381, Dec. 2017, doi: 10.1109 / JPROC. 2017.2768666.
[35] Y. Qiu, H. Zhang, K. Long and M. Guizani, "Subchannel Assignment and Power Allocation for Time-Varying Fog Radio Access Network With NOMA," in IEEE Transactions on Wireless Communications, vol. 20, no. 6, pp. 3685-3697, June 2021, doi: 10.1109 / TWC. 2021.3053004.
[36] H. Zhao, Q. Zheng, W. Zhang, B. Du and H. Li, "A Segment-Based Storage and Transcoding Trade-off Strategy for Multi-version VoD Systems in the Cloud," in IEEE Transactions on Multimedia, vol. 19, no. 1, pp. 149-159, Jan. 2017, doi: 10.1109/TMM.2016.2612123.
[37] J. G. Andrews, A. K. Gupta, and H. S. Dhillon. (Apr. 2016). A primer on cellular network analysis using stochastic geometry. [Online]. Available: http://arxiv.org/abs/1604.03183.
[38] Z. Zhang, Z. Ma, Y. Xiao, M. Xiao, G. K. Karagiannidis and P. Fan, "Non-Orthogonal Multiple Access for Cooperative Multicast Millimeter Wave Wireless Networks," in IEEE Journal on Selected Areas in Communications, vol. 35, no. 8, pp. 1794-1808, Aug. 2017.
[39] Mario Graf, Christian Timmerer, and Christopher Mueller. 2017. Towards Bandwidth Efficient Adaptive Streaming of Omnidirectional Video over HTTP: Design, Implementation, and Evaluation. In Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys'17). Association for Computing Machinery, New York, NY, USA, 261271.
[40] Ching-Ling Fan, Wen-Chih Lo, Yu-Tung Pai, and Cheng-Hsin Hsu. 2019. A Survey on 360 Video Streaming: Acquisition, Transmission, and Display. ACM Comput. Surv. 52, 4, Article 71 (September 2019), 36 pages.
[41] J. G. Andrews, A. K. Gupta, and H. S. Dhillon. (Apr. 2016). A primer on cellular network analysis using stochastic geometry. [Online]. Available: http://arxiv.org/abs/1604.03183
[42] Filar, J. and Vrieze, K. (2012). Competitive Markov Decision Processes. Springer Science & Business Media
[43] Kaiqing Zhang, Zhuoran Yang, Tamer Basar, Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms, https://arxiv.org/abs/1911.10635v2.
[44] Lowe, Ryan and Wu, Yi and Tamar, Aviv and Harb, Jean and Abbeel, Pieter and Mordatch, Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments, Igor, NIPS, 2017.
[45] D. Huang, P. Wang and D. Niyato, "A Dynamic Offloading Algorithm for Mobile Computing," in IEEE Transactions on Wireless Communications, vol. 11, no. 6, pp. 1991-1995, June 2012.
[46] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," in IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 4, pp. 974-983, 1 April 2015, doi: 10.1109/T-PDS.2014.2316834.
[47] C. Su, F. Ye, T. Liu, Y. Tian and Z. Han, "Computation Offloading in Hierarchical Multi-Access Edge Computing Based on Contract Theory and Bayesian Matching Game," in IEEE Transactions on Vehicular Technology, vol. 69, no. 11, pp. 13686-13701, Nov. 2020.
[48] Q. Zhang, H. Wang, Z. Feng and Z. Han, "Many-to-Many Matching-Theory-Based Dynamic Bandwidth Allocation for UAVs," in IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9995-10009, 15 June15, 2021.
[49] D. Wu, L. Zhou, Y. Cai, H. Chao and Y. Qian, "PhysicalSocial-Aware D2D Content Sharing Networks: A ProviderDemander Matching Game," in IEEE Transactions on Vehicular Technology, vol. 67, no. 8, pp. 7538-7549, Aug. 2018.
[50] X. Song, H. Li, M. Yuan and Y. Huang, "Coverage Performance Analysis of Wireless Caching Networks With Non-Orthogonal Multiple Access-Based Multicasting," in IEEE Access, vol. 7, pp. 164009-164020, 2019.
[51] Choset. Principles of Robot Motion: Theory, Algorithms, and Implementations. MIT Press, 2005.
[52] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo and Z. Wang, "Predicting Head Movement in Panoramic Video: A Deep Reinforcement Learning Approach," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 11, pp. 2693-2708, 1 Nov. 2019.
[53] M. Qiao, M. Xu, Z. Wang and A. Borji, "Viewport-Dependent Saliency Prediction in 360 Video," in IEEE Transactions on Multimedia, vol. 23, pp. 748-760, 2021.



Han Xiao received the B.E. degree in computer science from the Jinan University, in 2017. He is currently pursuing the Ph. D degree in the Network Architecture Research Center, Beijing University of Posts and Telecommunications (BUPT), advised by Prof. Changqiao Xu. His research interests include reinforcement learning, multimedia communications and panoramic video transmission.

Changqiao Xu (Senior Member, IEEE) 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.

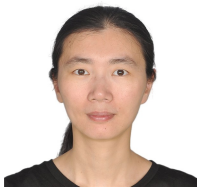
Zichen Feng received the B.E. degree from BUPT, in 2020, where he is currently pursuing the M.E. degree. His research interests include machine learning and multicast.



Renjie Ding received the B.E. degree from BUPT, and Queen Mary University of London, in 2021. And he is currently pursuing the M.E. degree in BUPT. His research interests include edge cache and panoramic video transmission.

Shujie Yang received the Ph.D. degree from the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China, in 2017. He is currently a lecturer with the State Key Laboratory of Networking and Switching Technology. His major research interests include wireless communications and multimedia communications.

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



Jie Liang is the expert in the China Telecom Internet and Application Technology Commission, senior expert in China Telecom IP network. She was once responsible for the technology research and network planning of China Telecom IP network and IPTV system, and has 20 years of working experience in the communication industry. Her research interests include video technology, multicast streaming and autonomous network.

Gabriel-Miro Muntean (Senior Member, IEEE) is a Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and co-Director 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.