A Video Cache Replacement Scheme based on Local Video Popularity and Video Size for MEC Servers

Pingshan Liu[1,2], Shaoxing Liu[2*], Zhangjing Cai[1], Dianjie Lu[3], Guimin Huang[2]
[1] Business School, Gulin University of Electronic Technology,Gulin,China
[e-mail: ps.liu@foxmail.com]
[2] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China
[e-mail: sxing.liu@foxmail.com]
[3] School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China
[e-mail: ludianjie@sdnu.edu.cn]
*Corresponding author: Shaoxing Liu

Received March 9, 2022; revised May 14, 2022; accepted August 1, 2022; published September 30, 2022

Abstract

With the mobile traffic in the network increases exponentially, multi-access edge computing (MEC) develops rapidly. MEC servers are deployed geo-distribution, which serve many mobile terminals locally to improve users' QoE (Quality of Experience). When the cache space of a MEC server is full, how to replace the cached videos is an important problem. The problem is also called the cache replacement problem, which becomes more complex due to the dynamic video popularity and the varied video sizes. Therefore, we proposed a new cache replacement scheme based on local video popularity and video size to solve the cache replacement problem of MEC servers. First, we built a local video popularity model, which is composed of a popularity rise model and a popularity attenuation model. Furthermore, the popularity attenuation model incorporates a frequency-dependent attenuation model and a frequency-independent attenuation model. Second, we formulated a utility based on local video popularity and video size. Moreover, the weights of local video popularity and video size were quantitatively analyzed by using the information entropy. Finally, we conducted extensive simulation experiments based on the proposed scheme and some compared schemes. The simulation results showed that our proposed scheme performs better than the compared schemes in terms of hit rate, average delay, and server load under different network configurations.

Keywords: mobile terminal, video utility, video popularity, cache replacement, multi-access edge computing

The research was supported by the national key research and development program of China (No. 2020YFF0305300), the National Natural Science Foundation (No. 61762029, No. U1811264, No. 61972237), the Shandong Provincial Natural Science Foundation of China (No. ZR2019MF017), Guangxi Key Laboratory of Trusted Software (No. kx201726).

http://doi.org/10.3837/tiis.2022.09.012 **ISSN : 1976-7277**

1. Introduction

With the increasing perfection of the streaming media technology, the video streaming media applications account for most of the total network traffic, and have become one of the main applications in the Internet. At present, the network video service has become one of the fastest growing businesses in the Internet industry [1]. At the same time, the number of mobile devices connected to the global fifth generation mobile communication system (5G) will soon exceed billions level [2], which is far faster than the growth rate of the 4-G era [3]. A wider range of intelligent devices (such as Internet of Things devices, 4K/8K video devices, and augmented reality and virtual reality devices) will continue to access the Internet [4], which will cause a large amount of traffic on the network access side. Cisco has also made relevant statistics on the network traffic load and predicted that mobile video traffic will account for 72% of the total data traffic by 2022 [5]. The existing network architecture and content delivery methods will have difficulty in meeting the ubiquitous and efficient network interconnection. A new architecture, Multi-access Edge Computing (MEC), is proposed. MEC deploys computing and storage resources to the network edge close to users, thereby providing users with high QoE (Quality of Experience), reducing the traffic of backbone network [6].

MEC server architecture can extend the content to the edge of the network. However, a lot of video transmissions on the network will still lead to high server load and large transmission delay. To optimize the allocation of server resources, some researchers studied resource allocation based on deep learning. Chen et al. [7] proposed a primary-prioritized recurrent deep reinforcement learning algorithm for dynamic spectrum access based on cognitive radio (CR) technology. Gu et al. [10] proposed a deep learning-based algorithm to solve the cache replacement problem of MEC servers in which the optimization problem is formulated as a constrained Markov decision process (MDP). Chen et al. [8] proposed a binary offloading scheme for edge computing, and introduced a deep reinforcement learning offloading model to acquire network resource allocation and optimally offloading decisions. For the user's view experience, it is obviously more important for video short-time delay response and transmission. The videos could be cached in the local server before the request, which can greatly improve the user's QoE. Therefore, more and more researchers focus on the cache prediction. For the proposed mobile edge computing (MEC) network, Lai et al. [9] proposed a cache prediction algorithm that combined the historical request data of the user with the channel between the relay and the user. And this algorithm solved the problem of content prediction and cache based on neural network and relay selection. Liang et al. [11] proposed a novel architecture that integrates mobile edge computing (MEC) in Social CCN (MeSoCCN) and proposed multi-head attention based on popularity prediction caching scheme in MeSoCCN. Shu et al. [12] proposed a Group Behavior and Popularity Prediction based Collaborative Caching (GPCC) scheme based on MEC architecture to reduce access delay. However, the limited cache capacity of a single base station will lead to serious server load during peak traffic. In order to alleviate this problem, many researchers believe that collaborative caching is of research significance. Ugwuanyi et al. [13] studied a collaborative caching scheme in the heterogeneous MEC networks. In order to minimize the total delay cost of all users requesting content in the MEC networks, proposed a cache management scheme. Considering the global service utility, collaborative service has gradually attracted the attention of researchers Yang et al. [14] introduced a new satellite-ground integrated collaborative caching network architecture based on MEC and studies the caching scheme. Wang et al. [15] considered an optimal content cache problem between Multi-access Edge Computing (MEC) servers to minimize the average content delivery delay limited by the

storage and computing power of each server, and proposed a MILP model and Mix-Cooperative (MixCo) cache scheme. Wang et al. [16] found that a massive number of cache strategies in MEC adopt LRU and LFU cache replacement strategies. Kurniawan et al [17] proposed the Modified-LRU algorithm. Modified-LRU takes an idea of SF-LRU, but it is simpler in the process of deleting files. Aimtongkham et al. [18] proposed a new LFU cache replacement scheme. Using SVM to classify video, LFU is applied to the actual replacement of a given new web object.

In this paper, an environment with only mobile users is considered, and we assume that user nodes are always on the move, at different speeds. The proximity principle is adopted for user nodes to access MEC server. However, the location uncertainty of the mobile node access point will cause the streaming video on demand service to switch servers many times. In addition, the mobility of devices will also cause rapid changes in local video popularity, which may make it difficult to predict global popularity. The requests of mobile users bring great challenges to local caching. Some classical LRU and LFU cache algorithms both are aimed at local videos, and have a wide range of applications and good performance. However, in the mobile environment, these cache algorithms do not perform well in alleviating the backhaul link load and improving the cache hit rate. It is well-known that the way to ensure the best users' QoE is to cache the video requested by the user in the local MEC server in advance. When the cache space of a MEC server is full, how to replace the cached video is an important problem, which is also called cache replacement problem. An optimal cache replacement scheme for MEC severs can effectively reduce the waiting delay, improve the cache hit rate, and reduce the backhaul link load. The previous studies on the cache replacement schemes for MEC servers failed to consider local video dynamic popularity and varied video size together. The local video dynamic popularity and the varied video size make the cache replacement problem more complex. Therefore， to design an optimal cache scheme for MEC servers, we proposed a new cache replacement scheme based on local video popularity and video size. The local video popularity is calculated based on the local popularity model, the construction of which is related to the access frequency. Furthermore, high quality and popular videos should be cached, and videos with high quality and low popularity should be replaced. To evaluate the importance of video popularity and video size in caching algorithm, we introduced information entropy to analyze the weight of the two.

The major contributions of this paper are shown as follows:

1) We proposed a new cache replacement algorithm based on the utilities of local videos. The core of the algorithm is a utility function combining local video popularity and video size. In order to effectively handle the dynamic of video popularity, a local video popularity model is designed, which is divided into popularity attenuation model and popularity rising model. In addition, considering the problem of a sudden reduction of video popularity caused by very few access frequencies in each time period, the attenuation model can be further divided into frequency dependent attenuation model and frequency independent attenuation model.

2) It is neither positive nor negative correlation between video popularity and video size. And the dynamicity of local video popularity makes the weight of video popularity and video size uncertain in the cache algorithm, Therefore, we introduced the information entropy to quantify the weight of the factors in the cache algorithm. The information entropy can reflect the influence probability of two factors in the cache.

3) According to the simulations, Under different conditions (the number of videos, the cache capacity and the number of requests), we conducted extensive simulations. The simulation results show that our proposed cache replacement scheme has better performance than LRU and LFU with thein terms of hit rate, waiting delay and server load.

The rest of the paper is organized as follows. In section 2, we review related work. Section 3 presents the system structure. Section 4 presents the problem formulation and the proposed scheme. We describe the simulations, and discuss the simulation results in section 5. We conclude the paper in section 6.

2. Related Work

In today's network, nodes can be divided into mobile nodes and fixed nodes. The geographic location of a stationary node is generally fixed. However, the geographic coordinates of mobile nodes are constantly changing, which also leads to the diversity of access by mobile devices. In order to better simulate the characteristics of mobile users in the network, Camp et al. [19] performed different models for different mobile nodes. For the mobility of nodes, there are mainly the following methods in the past research, for the Markov chain model with discrete variables with Markov characteristics [20], the random network model in the spatial dimension [22], and the residence time distribution model in the time dimension [21], and the Waypoint mode of random free movement of nodes [23]. In order to meet the application characteristics of short video services, and to further optimize cache and improve network performance, a modeling description suitable for user mobility is still needed. Li et al. [24] explored a realistic block scenario with femtocells deployed according to Poisson Point Process (PPP). And using the context information, a user motion trajectory fitting method based on *Lagrangian* interpolation is proposed. According to the slope of the trajectory polynomial and the distance between the user and the neighbor coverage cell, the transition probability of the user to the neighbor coverage cell is evaluated.

In terms of mobile video transmission, edge cache strategies have been proven to reduce the direct access to resources from the core network through the backhaul link, thereby ensuring low user latency [25]. In order to improve the efficiency of edge cache, some research work mainly focuses on the popularity of videos, and only caches the most popular video content that is most likely to be requested in the future. Ahlehagh et al. [26] initially proposed a cache scheme based on user preference profile (UPP). Yi et al. [27] proposed a pre-loading of videos that may be visited based on the social relationships of active users. These videos are cached on the viewer's mobile device in advance to achieve a smaller transmission delay. However, many users do not watch the video completely while watching the video, and therefore, it is easy to cause low utilization of the pre-cached video. Some works focus on energy efficiency and computational unloading. Chen et al. [28] studied a polling callback energy-saving offloading scheme, that is, the arrival time of data transmission and task processing time are asynchronous. To reduce the computational complexity of offloading computation under time varying channel conditions, they proposed a game-learning algorithm, that combines DDQN and distributed LMST with intermediate state transition (named DDQNL-IST). Zhou et al. [29] proposed a parameter adaptation-based ant colony optimization algorithm based on particle swarm optimization algorithm with the global optimization ability, which can improve the optimization ability and convergence and avoid to fall into local optimum. Zhao et. al [30] proposed a novel vibration amplitude spectrum imaging feature extraction method using continuous wavelet transform and image conversion is proposed, which can extract the image features with two-dimensional and eliminate the effect of handcrafted features. To improve the fog resource provisioning performance of mobile devices, Chen et al [31] proposed a learning-based mobile fog scheme with deep deterministic policy gradient (DDPG) algorithm. In addition, the DDPG algorithm is also used to solve the issue of state spaces explosion and learn an optimal offloading policy on distributed mobile fog

computing.

Many researches on content cache have made too simple assumptions. The authors of [32-33] all assumed that storage and bandwidth capacity is unlimited. In order to solve the problem of data prefetching and cache when the storage size of mobile users is limited. Liu et al. [34] designed a mobile perception utility function based on the user's movement probability and the popularity of video clips, and proposed a dynamic programming algorithm. Considering the rapidly changing edge network environment, Niyato et al. [35] measured the freshness of data by setting the life cycle of the data. When the life cycle of a certain data exceeds a threshold, the cache will be removed to maximize cache hit rate. Tran et al. [36] proposed a new content cache framework, which stores resources in a hierarchical cache method. They proposed a low-complexity heuristic cache algorithm to minimize the network cost of delivering content, thereby improving the quality of user experience. The authors of [37] proposed an online algorithm for content cache, which makes content cache and routing scheduling based on the characteristics of each request. Online algorithms can greatly improve the cache hit rate and reduce data backhaul traffic and response delay. The authors of [38-39] proposed a joint cache scheme for video streams in MEC networks. The joint cache scheme can not only improve the cache hit rate, but also reduce the consumption of the backhaul link and the initial access delay. However, these advanced pre-fetched video cache schemes have certain limitations. They do not consider the issue of video replacement in capacity. MEC can provide video storage and computing capabilities at the edge of the network. Optimizing the MEC cache content to maximize the cache hit rate is an important issue. There are static models and dynamic models in the existing literature to measure the popularity of content. At present, video popularity has also become an important factor for many researchers to apply cache strategies, but most research work on edge cache assumes that content popularity is static and uses an independent reference model. Paschos et al. [40] described the content request is generated based on an independent Poisson Process. Breslau et al. [41] observed a common video popularity model is the distribution of *Zipf* observed in the Web cache. However, this model does not reflect the popularity of videos in real time. The videos in the cache are constantly changing and the amount of access to each video is uncertain, which also makes the popularity of videos constantly changing.

3. System Structure

3.1 System overview

In this section, we introduce an edge service network architecture in mobile scenarios, which is composed of a cloud server, MEC servers, and mobile users. The cloud server can distribute content to the MEC servers. Each MEC server is associated with computation and networking capabilities and can provide a highly distributed computing environment close to the user end. In this work, we intend to user the servers for cache and computation and we assume that these MEC servers can transfer resources via backhaul links. In addition, the MEC servers can directly serve mobile users and improve efficiency using their local computing power and storage capacity. The location of the access point changes because of the movement of the user. For example, some people watch short videos while commuting, some people download videos while walking, and some people prefer to watch videos while traveling in cars or high-speed trains. When users move within the coverage area of the current MEC server, they can maintain a stable connection with that MEC server. On the other hand, when users move from the current MEC server coverage area to another MEC server coverage area, the devices need

to establish a connection with the new MEC server. Whether the videos being played by the users are cached in the new MEC server will directly influence the user experience. Some mobile users often experience playback freezes or errors when playing videos; this is because the device switches the server and sends a request for the video again. Moreover, a bad situation arises wherein the new MEC server cache misses the request. In case of this situation, the MEC server needs to obtain the content from the neighboring MEC servers or the cloud server through the backhaul link. The **Fig. 1** shows the system architecture.

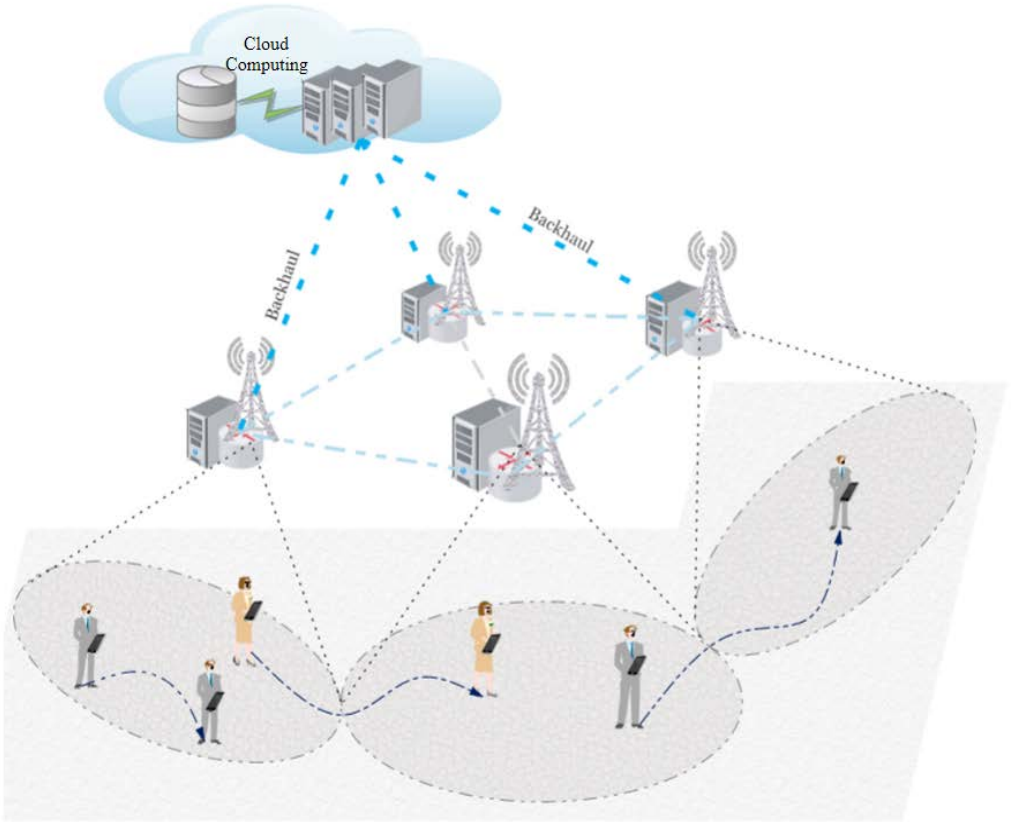

Fig. 1. System Architecture Model

We consider a set $\mathcal{M} = \{1,2,\ldots,M\}$ of MEC servers, a set $\mathcal{U} = \{1,2,\ldots,U\}$ of users and a set $\mathcal{F} = \{1,2,\ldots,F\}$ of videos. Let C_m and Bw_m indicate the capacity and upload bandwidth of the MEC server, respectively. Mobile users do not have storage capacity, and Bw_u represents the download bandwidth of user u $(u\epsilon\mathcal{U})$. The video size is denoted by $Size_f$. The video is transmitted in chunks by the streaming media technology. Therefore, the video content f $(f \in \mathcal{F})$ is divided into fragments of size S_{Seg}. **Table 1** presents the list of symbols used in this paper.

Table 1. List of symbols used in the paper

Symbol	Description
$\mathcal{M}$	A set of the MEC servers
$\mathcal{U}$	A set of the users
$\mathcal{F}$	A set of the videos
$Size_f$	The size of video f
Bw_u	The bandwidth of user u
Bw_m	The bandwidth of server m
C_m	The storage capacity of server m
S_{Seg}	The size of the video segment
$eval_f$	The evaluation parameter of the popularity model
P_f	The popularity of initial video f
pop_{f,un_freq}^{down}	The popularity of video f drops by the FIDA model
$pop_{f,freq}^{down}$	The popularity of video f drops by the FDA model
pop_f^{up}	The rise popularity of video f
P_f'	The popularity of local video f
$uFun_{f\epsilon\mathcal{F}}^{m\epsilon\mathcal{M}}$	The utility of video f in server m
E_j	The information entropy of factor j
w_j	The weight of factor j

3.2 Mobile Model

Nowadays, the habits of viewers are slowly shifting from the PC terminals to the mobile terminals in the Internet. Compared with the request of the PC terminal, there are many uncertain factors in the request of the mobile user. For example, the uncertainty of the moving path and the randomness of playing video. In addition, the movement of users is not always going in a certain direction or making a circular motion around a point. To reflect the movement trajectory of the user more realistically. We use *Gauss-Markov Mobility Model* to model the nodes. The nodes in the model are independent of each other. The movement of one node does not affect other nodes. Compared with ordinary random models, the speed and direction of node movement in *Gauss-Markov Mobility Model* is smooth, which can approximate the real mobile environment. As a result, the model can effectively eliminate the abrupt stop and abrupt steering problems encountered in the random walk process.

According to the movement regularity defined by *Gauss-Markov Mobility Model*, the next movement of the node needs to be calculated with reference to the speed, direction value of the previous movement. The calculation formula for the speed and direction of the node is as follows.

$$v_n = \alpha v_{n-1} + (1-\alpha)\bar{v} + \sqrt{1-\alpha^2} \cdot \theta_{n-1} \quad (1)$$

$$d_n = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{1-\alpha^2} \cdot \omega_{n-1} \quad (2)$$

In (1) and (2), $v_n$and d_n respectively represent the value of the speed and direction of the node at time n. $\bar{v}$ and $\bar{d}$ represent the average value of speed and direction respectively, and both θ_{n-1} and $\omega_{n-1}$are random variables conforming to Gauss distribution at time $n-1$. α represents a moving smoothness factor ($0 \leq \alpha \leq 1$). The degree of randomness in the mobility model can be changed by setting α. If $\alpha = 0$, node movement is completely random, and when $\alpha = 1$, node movement is a linear movement.

The coordinates of the nodes can be calculated by combining the previous coordinates, speed and direction values of the nodes. The calculation formula is as follows:

$$X_n = X_{n-1} + v_{n-1} \cdot cosd_{n-1} \quad (3)$$

$$Y_n = Y_{n-1} + v_{n-1} \cdot sind_{n-1} \quad (4)$$

In (3) and (4), X_n and Y_n respectively represent the x and y coordinates values of the node at time n.

4. Problem Formulation and Algorithm

This section mainly describes the popularity representation model, and formulates the cache replacement scheme.

4.1 Local Video Popularity Model

In related work, some researchers assumed that video popularity only satisfies the distribution of *Zipf*. But the distribution can only reflect a universal phenomenon. the popularity of the video should be constantly changing. For example, some videos are promoted through advertisements, poster notices, etc., before being officially released. After being released, the popularity of a video is affected by many factors, among which the access frequency of users is a critical factor. After a video is released, the access frequency generally goes through the following stages in the network: the access frequency a) reaches a peak from 0, b) drops rapidly from the peak to the trough, and c) fluctuates dynamically within a certain range or is always 0. It can be seen from the above that it is difficult to determine the dynamic popularity. We know that the most critical factor affecting popularity is the access frequency. If a video has been accessed frequently recently, its popularity must be high, and vice versa. Therefore, we design a local popularity model. Therefore, in this paper, we design a local popularity according to the video access frequency. And, the popularity model can be divided into the popularity attenuation model and the popularity rise model.

Popularity attenuation model: A decrease in the access frequency of a video will cause popularity attenuation. When the access frequencies of videos are very little or no in recent times, the popularity of the video drops rapidly. This is an automatic decline in popularity. At this time, the popularity decline value calculated using the access frequency may tend to be 0, which is unrealistic. Therefore, the popularity attenuation model need be divided into the frequency-dependent attenuation and the frequency-independent attenuation models. For convenience, we use the FDA and the FIDA to represent the frequency-dependent attenuation and the frequency-independent attenuation, respectively.

Popularity rise model: The dominant factor affecting the popularity increase is the access frequency of the users. For example, some videos are watched by many users after being released, and the popularity of these videos should meet the popularity rise model. In addition, the access frequency of a video gradually popularity increases with time, the video popularity should meet the popularity rise model in this case as well.

4.1.1 Evaluation Parameter of Popularity

In general, the increase or decrease in video popularity in each period is relative to the access frequency of the video in the previous period, and the current access frequency has a greater impact on the video popularity than the previous access frequency. Some videos, whose access frequencies were high in the previous period, "expire" as time passes making them no longer popular. Some videos were rarely viewed in previous periods, but may be accessed by massive users in the current period. These videos belong to popular videos. Therefore, calculating the video popularity by simply accumulating the frequency of each period would not yield accurate results. The access frequency in different periods has different effects on the video popularity. In addition, it is very valuable to judge whether the popularity meets the attenuation model or rise model. In this paper, we introduce a parameter $eval_f$, which can determine whether the video popularity meets the popularity rise model or the FDA model or the FIDA model. $eval_f$ is given by

$$eval_f = \sum_{i=1}^{n} eval_f^{\Delta freq_i} \tag{5}$$

where $eval_f^{\Delta freq_n}$ represents the influence of the access frequency in adjacent periods on the video popularity, n represents the number of period intervals, f denotes the video number. $eval_f^{\Delta freq_n}$ is given by

$$eval_f^{\Delta freq_n} = \frac{freq_{T_n} - freq_{T_{n-1}}}{\rho \cdot 2^n} \tag{6}$$

where $freq_{T_n}$ and $freq_{T_{n-1}}$ represent the access frequencies of periods T_n and T_{n-1}, respectively. We assume that the access frequency of the initial period T_0 is 0， ρ represent the balance coefficient, and is used to balance the influence of excessive frequency on the exponential function.

If $eval_f > 0$, the popularity rise model is satisfied; if $eval_f < 0$, the popularity attenuation model is satisfied; and, if $\tau < eval_f < 0$, the FDA model is satisfied. The smaller the value of $eval_f$, the lower the current access frequency of the video. If $eval_f < \tau$ (τ represents a threshold and is a negative number), it means that the video has not been accessed recently, and the FIDA model is satisfied. A special case of $eval_f = 0$, means that the access frequency of the video is the same in all periods. If the access frequency of each period is 0, the FIDA model is satisfied; otherwise, it conforms to the FDA model.

4.1.2 Popularity Attenuation Model

If $eval_f < 0$, it means that the video popularity is attenuated. Popularity attenuation has two different cases. Case 1: A video has not been accessed for a long time and its popularity typically decreases rapidly with time. Case 2: A video has access frequency in different periods, but the video popularity is decrease. Therefore, to calculate the popularity under different situations, we propose two attenuation models, namely, the FIDA model and the FDA model.

A. FIDA Model

The authors of [42] mentioned that, according to the Newton's law of cooling, an object with temperature higher than that of its surrounding environment will transfer heat to the surrounding medium, and the temperature of this object will decrease over time until it becomes equivalent to the ambient temperature. The mathematical formula for Newton's law of cooling is a differential equation. The popularity of a video that has not been accessed for a long time decreases rapidly over time. Therefore, the problem of video popularity attenuation

conforms to a linear model. If $eval_f < \tau$, the function relationship between video popularity and time can be established using the Newton's law, thereby obtaining the FIDA model. This model can reflect changes in popularity over time, regardless of the frequency; it is given by

$$-\frac{1}{k}\Delta T = \frac{pop_{f,unfreq}^{down} - P_{low}}{P_f - P_{low}} \tag{7}$$

$$pop_{f,un\_freq}^{down} = P_{low} + (P_f - P_{low}) \cdot e^{-\frac{1}{k}\Delta T} \tag{8}$$

where pop_{f,un_freq}^{down} represents the video popularity after attenuation; P_{low} represents the lowest popularity; (the minimum value of the initial popularity P_f can be set as the lowest popularity); k represents the correlation coefficient; and ΔT represents the interval of the period.

B. FDA Model

If $\tau < eval_f < 0$, the FDA model is satisfied. The FIDA model can better represent the rapid decrease in popularity of videos that have not been watched in the recent period. However, most videos have an access frequency in each period, and the frequency is different. The FIDA model could not reflect this popularity attenuation. Therefore, we designed FDA model, which reflects the true attenuation of popularity of videos with access frequencies. In order to describe the impact of access frequency on popular in different periods, the model relates the access frequency in different periods with an exponential function, and then obtains an attenuation value of popularity, it is given by

$$pop_{f,freq}^{down} = \frac{\sum_{i=1}^{n} p_f^{T_n}}{\sum_{t=1}^{T_{total}} freq_{T_t}} \tag{9}$$

where $pop_{f,freq}^{down}$ denotes the decreasing popularity of video f, T_{total} denotes the total number of periods, and $p_f^{T_n}$ denotes the influence of access frequency on video popularity in period T_n.

$$p_f^{T_n} = \frac{freq_{T_n}}{2^{T_{total} - T_n}} \tag{10}$$

where $freq_{T_n}$ denotes the frequency in period T_n.

4.1.3 Popularity Rise Model

If $eval_f > 0$, it means that the video popularity increases. The most important factor affecting the rise of video popularity is the amount of video playback. To calculate local video popularity, many previous studies only add 1 to the total clicks of the video when a user watches the video once. Then, the local video popularity is calculated based on the total access frequencies. However, the above method is unreasonable because the access frequencies in different periods have different effects on the video popularity. For example, some videos that were rarely watched in the previous period, but be widely watched with coincidence by many users in the current period. At this moment, the popularity of the videos may be higher than that of some videos with high access frequencies in the past. Therefore, the recent access frequency is more important than in the past. To solve the problem of rise popularity, we design a popularity rise model. The above-mentioned FDA model shows that the decrease in frequency is negative feedback on popularity. Similarly, increasing frequency can be used as positive feedback on popularity. The popularity rise model is analogous to the frequency attenuation model, and is formulated as follows:

$$pop_f^{up} = \frac{\sum_{i=1}^{n} p_f^{T_n}}{\sum_{j=1}^{T_{total}} freq_{T_j}} \tag{11}$$

$$p_f^{T_n} = \frac{freq_{T_n}}{2^{T_{total}-T_n}} \tag{12}$$

where pop_f^{up} denotes the increase in popularity. The other symbols were defined already in (9) and (10).

If the video popularity is attenuated, the video will not be affected by the popularity rise model, and vice versa. The popularity attenuation model and popularity rise model show that local video popularity P_f' satisfies a piecewise function:

$$P_f' = \begin{cases} pop_{f,un\_freq}^{down}, & eval_f < \tau \\ P_f - pop_{f,freq}^{down}, & \tau < eval_f < 0 \\ P_f + pop_f^{up}, & eval_f > 0 \end{cases} \tag{13}$$

where $eval_f < \tau$ means that the video has not been watched for a long time and belongs to a type of FIDA model; $\tau < eval_f < 0$ means that video popularity meets the FDA model; and $eval_f > 0$ means that the popularity rise model is satisfied. In addition, if $eval_f < 0$, two cases arise. The first is that the access frequency of each period is not 0, and is the same. The second is that the access frequency of each period is 0. The former can be classified as satisfying the FAR model, and the latter can be classified as satisfying the FIDA model.

4.2 Cache Replacement Algorithm

In this paper, we propose a cache replacement scheme for local MEC server, which is based on local video popularity and video size. Higher cache local hit rate can effectively reduce the waiting delay of user and backhaul link load. During off-peak traffic periods, the delay caused by the backhaul link may be tolerated. However, during peak traffic periods, excessive requests that need to obtain resources through the backhaul link are not allowed. A high backhaul link traffic causes the transmission channel to be blocked, leading to a long waiting time for users. The local cache the video can alleviate the backhaul link traffic well. Because the bandwidth and storage capacity of the MEC servers are valuable resources, not all videos can be cached. If the storage capacity of MEC servers is insufficient, it is important to cache as many videos as possible that may be viewed in the future. Therefore, our proposed cache replacement scheme considers two video factors that are the local video popularity and video size. And, we introduce a utility function $uFun_{f\epsilon\mathcal{F}}^{m\epsilon\mathcal{M}}$, which denotes the value of video f in edge server m. The video with the minimum $uFun_{f\epsilon\mathcal{F}}^{m\epsilon\mathcal{M}}$ in the cache is replaced first. Note that the requested videos cannot be replaced. We use $isReqed_m^f$ to denote the status of the video. $isReqed_m^f = 1$ indicates that the current videos are being requested and cannot be replaced. $isReqed_m^f = 0$ indicates that the videos can be replaced. $uFun_{f\epsilon\mathcal{F}}^{m\epsilon\mathcal{M}}$ is calculated as follows.

$$uFun_{f\epsilon\mathcal{F}}^{m\epsilon\mathcal{M}} = \begin{cases} w_1 P_f' + w_2 Size_f, & isReqed_m^f = 0 \\ max, & isReqed_m^f = 1 \end{cases} \tag{14}$$

whereP_f' represents local video popularity, $Size_f$ represents the video size, max represents a sufficiently large value, that is greater than the largest $uFun_{f\epsilon\mathcal{F}}^{m\epsilon\mathcal{M}}$, and ω_1 and ω_2 are weighting factors ($\omega_1 + \omega_2 = 1$).

In the network service system, the MEC server provides mobile users with high-quality services using our proposed cache algorithm, which is not unlimited.

$$Constraint\ to:$$
$$\sum_{f=1}^{F} x_f^m \cdot Size_f \leq C_m \ , \forall m \in \mathcal{M} \quad (14\ a)$$
$$\sum_{u=1}^{U} upload_u \cdot x_f^m \leq Bw_m \ , \forall m \in \mathcal{M} \quad (14\ b)$$
$$x_f^m = \{0,1\}, \forall m \in \mathcal{M}, f \in \mathcal{F} \quad (14\ c)$$

Where u denotes the user number. The constraints applied to the problem are described as follows: Constraint (14 a) guarantees the storage capacity of the server. Constraint (14 b) guarantees the bandwidth capacity of the server. Constraint (14 c) represents the possible value of the cache result, indicates whether the videos are cached by the server m. The pseudo code of the proposed cache algorithm is presented in *Algorithm 1*.

Local video popularity and video size are two decisive factors for our proposed cache algorithm, and the weights of the two factors cannot be set arbitrarily. For example, some high-quality videos with low popularity are cached, a mass of user requests will not be hit videos in the caching, and the cache space will be wasted. It is obvious that high-quality videos, with high popularity should be cached. In contrast, high-quality and unpopular videos should be replaced. The high popularity of a video means that it is more likely to be visited in the future. High-quality videos take up more storage space, and require longer times to be transmitted from other MEC servers or remote cloud server. Hence, the weights of local popularity and the video size need to be further analyzed in the cache replacement algorithm. In this paper, we introduce the information entropy for evaluating the weights. Due to $0 < P_f' < 1$, video popularity need not be standardized. The normalization formula of $Size_f$ is $Size_f' = \frac{Size_f - Size_{min}}{Size_{max} - Size_{min}}$, $Size_{max}$ denotes maximum video size, $Size_{min}$ denotes minimum video size. We let $g_{f,1}$ denotes the P_f', and $g_{f,2}$ denotes $Size_f'$ in the formula given below.

Algorithm 1

Initialize the moving area and the initial coordinates and moving speed of the nodes, the coordinates and coverage radius of MEC servers
Initialize P_f, $Size_f$, Bw_m, and C_m
If (the available capacity of MEC is enough)
    Cache directly
else
    Normalize the video size
    If (period == 1)
        local video popularity P_f' comes from the initial popularity
        Calculate E_j , ω_j according to formula (15) and (16) respectively
        Calculate $uFun_{f\epsilon\mathcal{F}}^{m\epsilon\mathcal{M}}$ according to formula (14)
        Sort $uFun_{f\epsilon\mathcal{F}}^{m\epsilon\mathcal{M}}$ of all videos in descending order
        Replace the videos with the minimum $uFun_{f\epsilon\mathcal{F}}^{m\epsilon\mathcal{M}}$ in turn
    Else

```
14        Calculate the popularity evaluation parameter eval_f
15        If (eval_f < τ)
16          Calculate P'_f according to formula (8)
17        Else If (τ < eval_f < 0)
18          Calculate P'_f according to formula (9)
19        Else If (eval_f > 0)
20          Calculate P'_f according to formula (11)
21        Else
22          If (The access frequency of the videos in each period is 0)
23            Perform step (16)
24          Else
25            Perform step (18)
26          End If
27        End If
28        Perform step (9) to (12)
29      End If
30    End If
```

The information entropy is defined to measure the amount of information generated by an event. A high- probability event occurs frequently, but the amount of information generated by this event is less. On the contrary, a low probability event occurs rarely, but the amount of information generated by this event massive. For example, the event of rain on the next day would generate less information, whereas the event of a large earthquake on the next day would generate large amount of information. The popularity and size of a video are regarded as two pieces of information in the cache process. Therefore, it is feasible to evaluate the importance of factors through the information entropy. We take $g_{f,1}$ and $g_{f,2}$ as local popularity and normalized size of the video, respectively. The information entropy E_j can be calculated based on the definition of information entropy in the information theory:

$$E_{j\epsilon\{1,2\}} = -\frac{1}{ln\mathcal{F}}\sum_{f=1}^{\mathcal{F}} P_{f,j}\, ln_{P_{f,j}} \tag{15}$$

where $P_{f,j}$ represents the information probability of index j of video f, and $P_{f,j} = \frac{g_{f,j}}{\sum_{f=1}^{\mathcal{F}} g_{f,j}}, j \in \{1,2\}$.

In the definition of information entropy, the greater the entropy of information, the smaller the probability of occurrence. However, in the cache, the greater the entropy of information, the more important it is, and vice versa. Therefore, the weight ω_j of local video popularity and video size can be calculated as follows:

$$\omega_{j\epsilon\{1,2\}} = \frac{1-E_j}{2-(E_1+E_2)} \tag{16}$$

where ω_1 denotes the weight of local video popularity, and ω_2 denotes the weight of video size.

4.3 The computational complexity of the algorithm

The time complexity of the proposed algorithm is $O(KN^2 + (logN + AT + B)N)$, as shown in *Algorithm 1*. We let N denote the number of videos in the local cache, let T denote the number of running cycles, and let K denote the number of weight factors. A and B are constant coefficients.

Proof: In *Algorithm 1*, the first step is in line 6, which takes time $O(N)$. The second step is in line 9 or from line 14 to line 26. The time complexity of line 9 is $O(1)$. From line 14 to line 26, the program can be divided into two steps. The time complexity of the first step is $O((T-1)N))$ in line 14. The time complexity of the second step is $O(TN)$ from line 15 to 26. The third step is in line 28, the program on lines 9 to 12 is executed. In line 9, the time complexity of calculating information entropy is $O(KN^2)$, and the time complexity of calculating the weights is $O(K)$. The time complexity of line 10 is $O(N)$. In line 11, the time complexity of the merge sort is $O(NlogN)$. To sum up, the time complexity of our algorithm is $O(KN^2 + N(logN + AT + B))$.

5. Performance Evaluation

In this section, we demonstrate the performance of our scheme through simulation. First, we introduce the set of simulation. Then, we introduce the metrics of the simulation. Finally, our scheme is compared with other classical cache replacement algorithms.

5.1 Simulation Settings

The parameter settings are shown in **Table 2**. In the system architecture, a Cloud server, three MEC servers and 3000 mobile nodes are deployed. The upload bandwidth of MEC is set to 200 Mbps. In order to simulate the realistic network scenarios, the nodes join the system randomly, and the movement of the nodes follow the Gaussian Markov movement model. The simulation experiment adopts a library of 400 videos with their popularity following the *Zipf* distribution with parameter $\alpha = 0.5$. The videos are selected with three versions, namely $l_1 = 480p$, $l_2 = 480p$ and $l_3 = 480p$. The bandwidth required for 480p is 1 Mbps. 720p requires a bandwidth of 2 Mbps. The bandwidth required for 1080p is 4 Mbps. In a real network environment, there are long movies and short videos in the cache. In order to facilitate the statistics of experimental data, the duration of the short video is set to 5 min, and the duration of the long movie is set to 1 h. The download bandwidth of nodes should to meet the actual mobile devices. The authors of [43] stated that the researchers have calculated the average bit rate in Twitch and learned that it is about 2 Mbps. We assume the bandwidth of mobile nodes is randomly distributed in a range [1,4] Mbps. The access delay of the video blocks received by users in different scenarios follows a uniform distribution. The delay is in a range of [5, 10] ms from cloud server, [1, 2.5] ms from neighboring MEC servers, [0.25, 0.5] ms when served from the home MEC server.

In order to simulate the sudden interruption of the connection between user nodes and the server, some nodes randomly choose to exit the network during the experiment. In addition, when user nodes run away from the coverage of the current MEC server, the current user node will disconnect from the server. Therefore, in order to save resources, we use the heartbeat packet to judge whether user nodes are connected to the current MEC server. In the simulation experiment, when the mobile nodes do not transmit data, these nodes are set to send a heartbeat packet to the local server every 15 seconds. The heartbeat packets received by the server indicates that these nodes have no fault, and vice versa. If mobile nodes transmit data with the

MEC server, it does not need to send a heartbeat signal.

In order to better evaluate the performance of our scheme, the algorithm will be compared with the classic LRU and LFU cache algorithms. The indicators for comparison are the Hit-Rate, the Cloud Server load, the transmission delay distribution of users and the playback continuity index.

Table 2. List of experimental parameters

Parameter	Definition	Value
M	The number of MEC servers	3
F	The number of videos	400
U	The number of mobile users	3000
Bw_m	The bandwidth of MEC server	200 Mbps
$delay_1$	The delay of local MEC server feedback	[0.25,0.5] ms
$delay_2$	The delay of neighboring MEC server feedback	[1,2.5] ms
$delay_3$	The delay of cloud server feedback	[5,10] ms
α	The parameter of the *Zipf* distribution	0.5
$upBand\_l_1$	The upload bandwidth required for 480 p	1 Mbps
$upBand\_l_2$	The upload bandwidth required for 720 p	2 Mbps
$upBand\_l_3$	The upload bandwidth required for 1080 p	4 Mbps

5.2 Evaluation of Weights

The weights represent the importance of local video popularity and video size. We set C_m to 30000 Mb, the number of video requests to 5000, and the number of videos to 400. And assume that the highest video popularity is 0.99 and the lowest is 0.01. We counted 100 sets of the weights. As is shown in **Fig. 2** and **Fig. 3**, we see that the weight of video popularity is [0.68,0.84], and the weight range of video size is [0.18,0.32]. The weight of video popularity is mainly distributed around 0.78 and the weight of video size is mainly distributed around 0.22.

5.3 Transmission Delay Distribution of The Users

The transmission delay represents the request response time of the video. Larger transmission delay leads to worse user QoE. To evaluate the performance of our scheme, LFU and LRU. We set C_m to 30000 Mb, the number of videos to 400, and the number of requests to 5000. In **Fig. 4**, (a), (b) and (c) shown the transmission delay distribution of our algorithm, LRU and LFU respectively. Compared to **Fig. 4** (b) and (c), the delay of nodes distributed in [1,2.5] ms and [5,10] ms are the least in **Fig. 4** (a). The delay distributed in [5,10] ms are the most in **Fig. 4** (c), and the delay distributed in [1,2.5] ms are the most in **Fig. 4** (b). The delay generated through our scheme is most distributed in [0.25,0.5] ms, which shows that the performance of our algorithm is better than the LRU and LFU in terms of transmission delay.

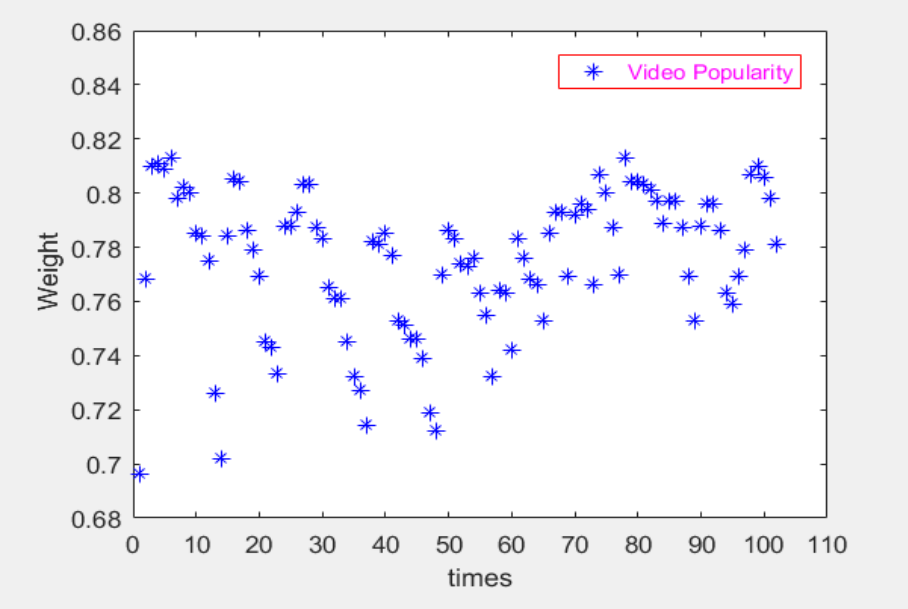

Fig. 2. The weight of the video popularity

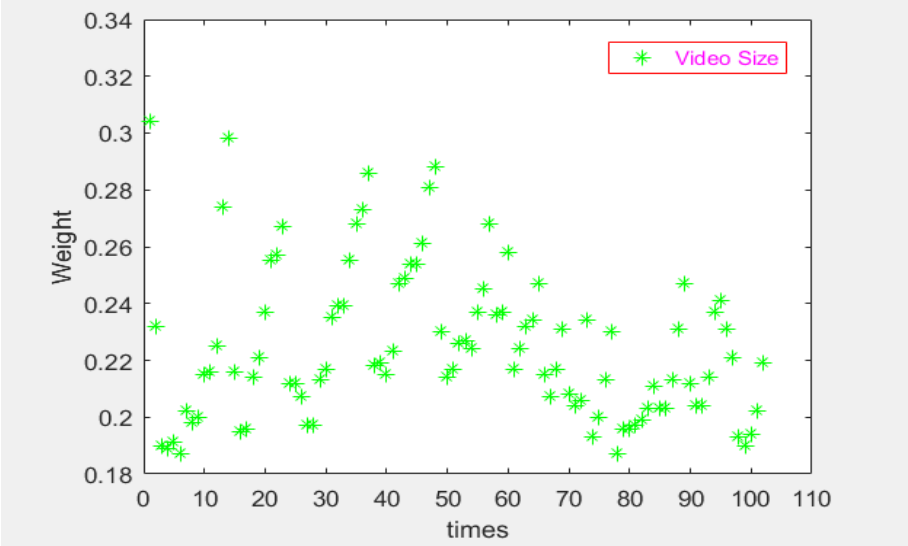

Fig. 3. The weight of the video size

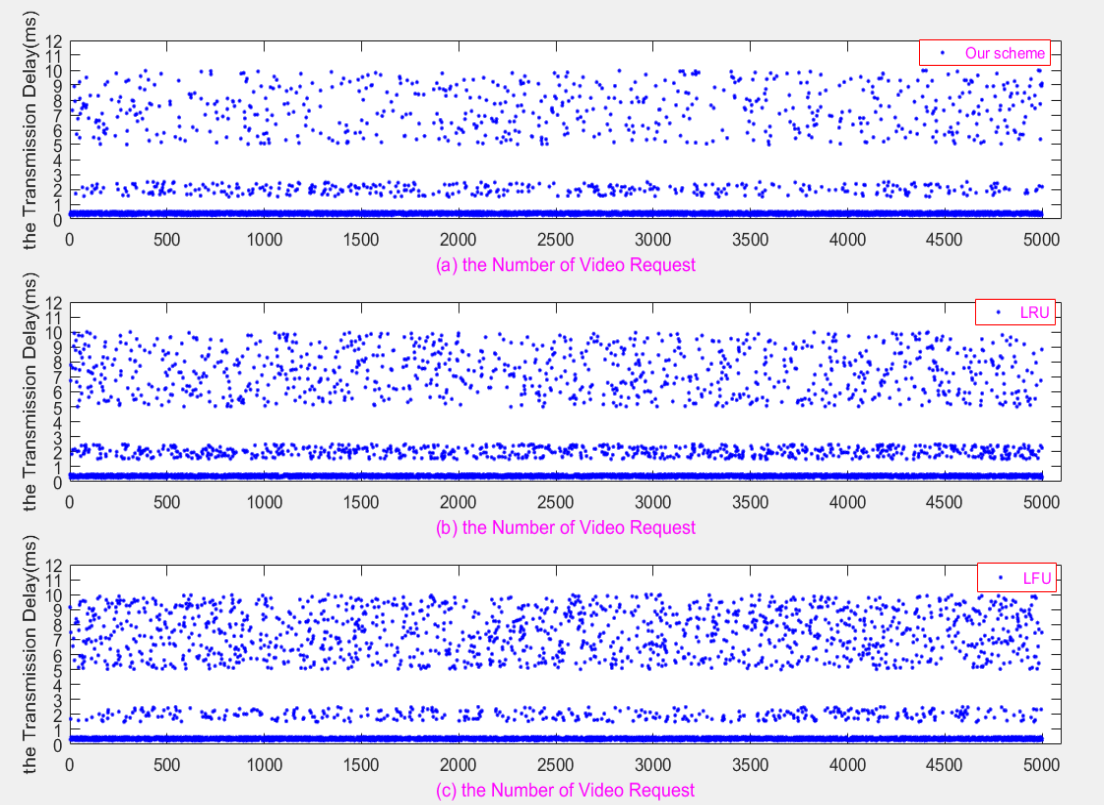

Fig. 4. The transmission delay distribution of the user requests

5.4 Hit-Rate

The hit-rate is defined as the ratio of video requests found to be stored in local cache to the total video requests. And the hit rate is the best way to reflect the performance of the schemes in the cache. The performance of our scheme is compared with LRU and LFU algorithms under different MEC server capacities. In order to ensure the accuracy of the experimental data, we have counted the experimental data of 15 periods, and the data in each period are the average value after repeated simulations for 5 experiments. As shown in **Fig. 5**, With different capacities, the overall hit rate of our scheme is higher than the LRU and LFU algorithms. The C_m is set to 10000 Mb, the hit rate of our scheme and LFU algorithm reaches the highest in the 7th period, which is 0.65 and 0.48 respectively. After in the 7th period, the hit rate of our scheme is always close to 0.59, while the hit rate of LFU is about 0.48. Before in the 7th period, the LFU algorithm has a serious jitter in the hit rate. This is because the MEC capacity is small and some outdated videos with high access frequency will exist for a long time. The highest hit rate of LRU is 0.41 in the 8th period. The hit rate of LRU decreased rapidly after the 8th period, this is because that MEC capacity is too small and frequent cache replacement leads to the elimination of some popular videos. C_m = 20000 Mb, the hit rate of our scheme reaches 0.74 in the 7th period, and is about 0.69 after. The hit rate of our scheme is about 0.17 higher than LFU and about 0.13 higher than LRU. C_m = 30000 Mb, the hit rate of of our scheme is about 0.18 higher than LFU and about 0.20 higher than LRU. C_m = 50000 Mb, many videos can be cached. Therefore, after the 6th period, the hit rates of both LRU and LFU algorithms are higher than 0.60. And the hit rate of our scheme exceeded 0.60 in the third period. After

the sixth period, the hit rate of our scheme is about 0.23 higher than LFU and about 0.20 higher than LRU.

Obviously, with the passage of time, the number of videos on the Internet is increasing. **Fig. 6** showed the MEC cache hit rate under different video numbers. We set C_m to 30000 Mb in the experiment. In addition, we extended the running time of the simulation experiment, and added 50 videos every 7 periods. The experimental data were counted once with an interval of 7 periods. As can be seen from **Fig. 6**, when the MEC storage capacity is certain, the hit rate of LRU and LFU is lower and the decline speed is faster than our scheme. When the number of videos is 400, the hit rate under our scheme is 0.82, while the hit rates of LRU and LFU are 0.65 and 0.63, respectively. When the number of videos reaches 800, the hit rate under our scheme is 0.67, while the hit rates of LRU and LFU are 0.42 and 0.35, respectively. Therefore, as the number of videos increases, the performance of our scheme is better than that of LRU and LFU.

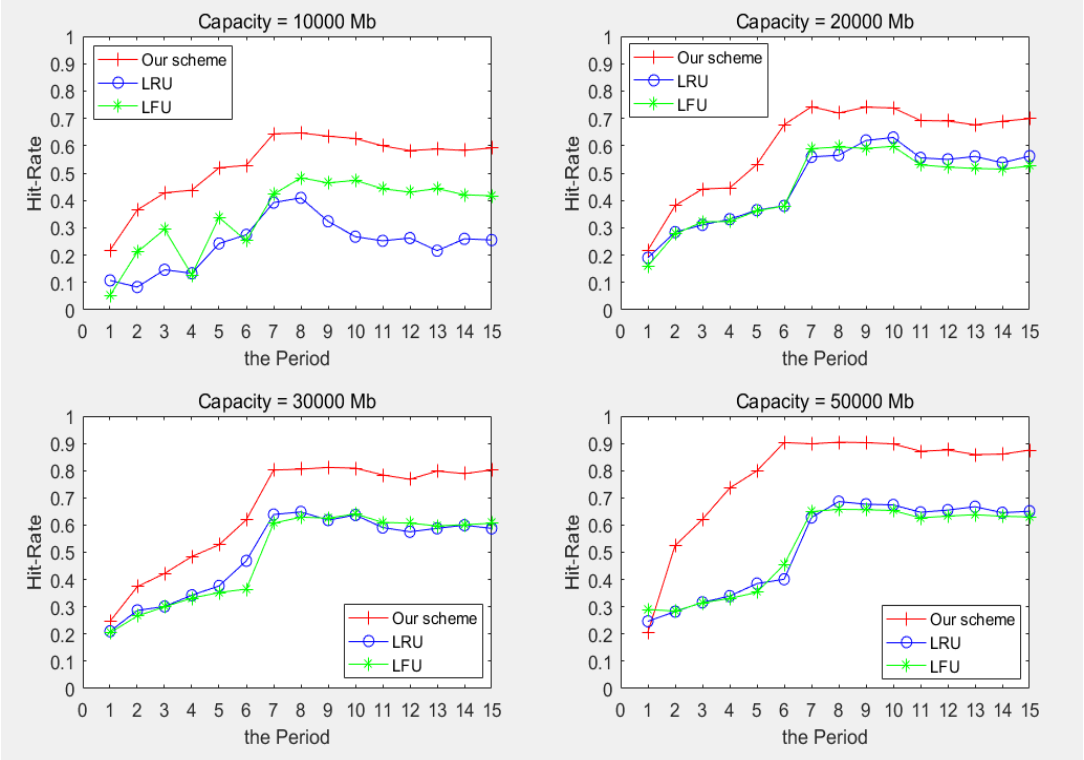

Fig. 5. The hit rate of with different the MEC server capacity.

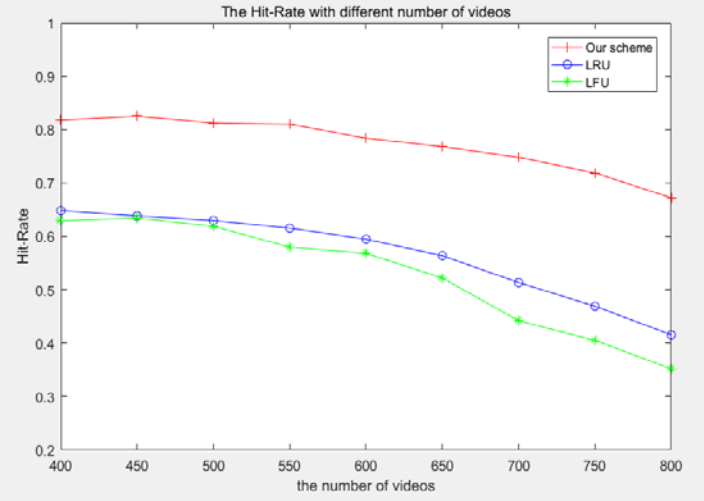

Fig. 6. The hit rate of with different number of videos

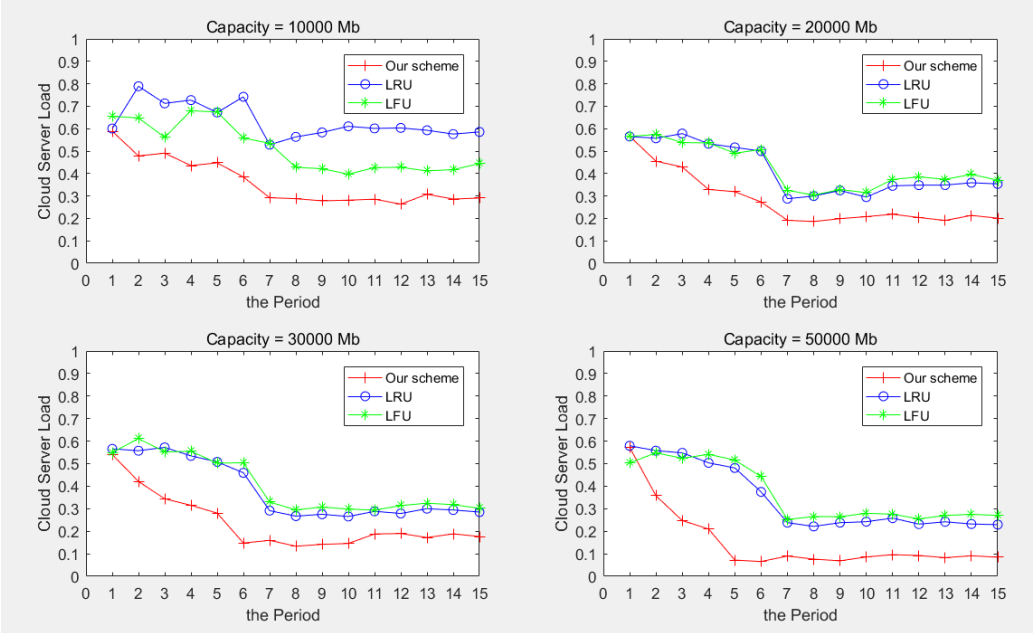

Fig. 7. Cloud server load with different the MEC server capacity.

5.5 Cloud Server Load

The Cloud server load is defined as the strain of user requests on the server. We assume that the cloud server has a maximum load of 1, at which point all requests are processed by the cloud server. We do extensive experiments and count the cloud server loads with three cache schemes in **Fig. 7**. the MEC server capacity is smaller, fewer videos will be cached and videos will be replaced frequently. With the MEC server capacity increases, the load decreases significantly. C_m =10000 Mb: The load of LRU and LFU algorithms is about 0.6 and 0.42 respectively in the 7th period. The load caused by our scheme is about 0.29, which is lower than LRU and LFU. In addition. C_m =20000 Mb: the cloud server load finally stabilizes at about 0.2 under our scheme, which is 0.14 and 0.17 lower than LRU and LFU respectively.

C_m is set to 30000 Mb or 50000 Mb, the server load with our scheme is about 0.17 and 0.09 respectively, about 0.28 and 0.23 with LRU respectively, and about 0.31 and 0.27 with the LFU respectively in the 7th period. Obviously, our scheme outstrips LRU and LFU algorithms at any moment.

In order to evaluate the performance of our scheme and other algorithms with different requests, we set C_m to 30000 Mb. **Fig. 8** depicted the load of the cloud server under different number of requests. The experiment period is set to 10. The experimental data is counted from the 7th cycle. It can be seen intuitively from **Fig. 8** that the performance our scheme outperforms the other two algorithms in the system. The number of requests from 1000 to 5000, the performance of the three algorithms keeps stable, but our scheme is compared with LRU and LFU, the cloud server load is lower. When the number of requests exceeds 6000, the load of the three algorithms gradually increases, the load increase rate of our scheme is slower than LRU and LFU algorithms.

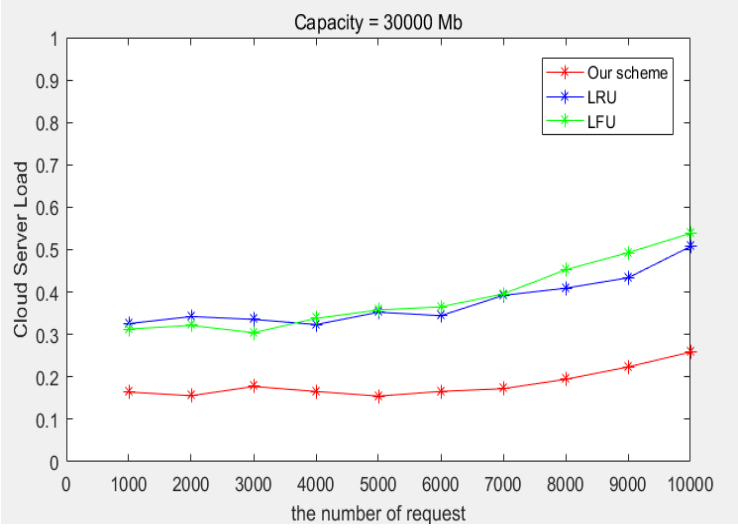

Fig. 8. Cloud server load with different number of requests

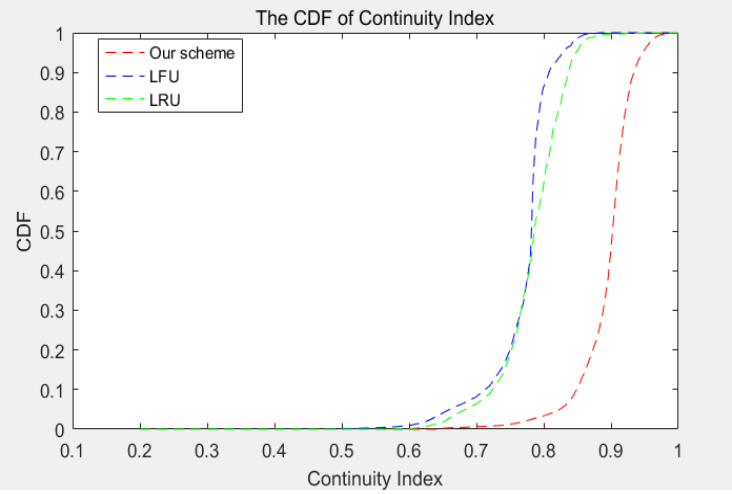

Fig. 9. The CDF of continuity index (C_m = 30000 Mb)

5.6 Continuity Index

The Continuity index indicates the smoothness of video playback. The calculation of the continuity index is the proportion of the number of video segments received to the total number of video segment requested by the nodes. If the continuity index is close to 1, it means viewing experience is better. As is shown in **Fig. 9**, We can observe that around 65% of the peers in the system with our scheme can have a continuity index over 0.9. However, while around 65% of the peers in the system with LRU and LFU algorithms can have a continuity index between 0.78 and 0.92, between 0.78 and 0.88 respectively. Obviously, our scheme can achieve the best performance among the three schemes.

6. Conclusion

In this paper, we have studied the cache replacement problem of MEC servers. We first constructed a network service architecture in a mobile environment. Then, we proposed a new cache replacement scheme based on the utilities of local videos, which comprehensively considers local video popularity and video size. To effectively solve the problem of dynamic popularity, we built a local video popularity model, which further is divided into a popularity attenuation model and a popularity rise model. Furthermore, considering the problem of a sudden reduction of video popularity caused by very few access frequencies in each time period, the attenuation model could be further divided into frequency dependent attenuation model and frequency independent attenuation model. In addition, we introduced information entropy to quantitatively analyze the weights involved in video utility. Therefore, our proposed scheme can perform well based on the local information of the MEC servers, which can integrate the local video popularity and video size into a video utility with information entropy. Finally, the extensive simulation results have been presented to demonstrate the performance of the proposed scheme.

In the future, we would like to extend our work in the interaction of MEC servers, multi-layer cache systems and reconstruct formulas and algorithms.

References

[1] Q. Zhang, W. Shi, H. Zhong, “Firework: data processing and sharing for hybrid cloud-edge analytics,” *IEEE Trans. Parallel Distrib. Syst*., 29(9), 2004–2017, 2018. Article (CrossRef Link) .

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper. Article (CrossRef Link).

[3] M. Yan, C. A. Chan, W. Li, “Network energy consumption assessment of conventional mobile services and over-the-top instant messaging applications,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3168–3180, 2016. Article (CrossRef Link).

[4] M. Yan, C. A. Chan, W. Li, L. Lei, A. F. Gygax, and C. L. I, “Assessing the energy consumption of proactive mobile edge caching in wireless networks,” *IEEE Access*, vol. 7, pp. 104394–104404, 2019. Article (CrossRef Link).

[5] Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021, 2017. Article (CrossRef Link).

[6] J. Sung, M. Kim, K. Lim, and J. K. K. Rhee, “Efficient cache placement strategy in two-tier wireless content delivery network,” *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1163–1174, 2016. Article (CrossRef Link).

[7] M. Chen, A. Liu, W. Liu, K. Ota, M. Dong, and N. N. Xiong, "RDRL: A Recurrent Deep Reinforcement Learning Scheme for Dynamic Spectrum Access in Reconfigurable Wireless Networks," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 2, pp. 364-376, 1 March-April 2022. Article (CrossRef Link).
[8] M. Chen, W. Liu, T. Wang, A. Liu, and Z. Zeng, "Edge intelligence computing for mobile augmented reality with deep reinforcement learning approach," *Comput. Netw*, vol. 195, pp. 108186, 2021. Article (CrossRef Link).
[9] S. Lai, R. Zhao, Y. Wang, "Content popularity prediction for cache-enabled wireless B5G networks," *EURASIP J. Adv. Signal Process*., 2021, 69, 2021. Article (CrossRef Link).
[10] J. Gu, W. Wang, A. Huang, H. Shan, and Z. Zhang, "Distributed cache replacement for caching-enable base stations in cellular networks," in *Proc. of 2014 IEEE International Conference on Communications (ICC)*, pp. 2648-2653, 2014. Article (CrossRef Link).
[11] J. Liang, D. L. Zhu, H. T. Liu, et. al, "Multi-Head Attention Based Popularity Prediction Caching in Social Content-Centric Networking with Mobile Edge Computing," *IEEE Communications Letters*, vol. 25, no. 2, pp. 508-512, Feb. 2021. Article (CrossRef Link).
[12] P. Shu, Q. Du, "Group Behavior-Based Collaborative Caching for Mobile Edge Computing," in *Proc. of 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 2441-2447, 2020. Article (CrossRef Link).
[13] Z. Sang, S. Guo, Y. Wang, "Collaborative Video Cache Management Strategy in Mobile Edge Computing," in *Proc. of 2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6, 2021. Article (CrossRef Link).
[14] Y. Li, X. Kong, Y. Qi, C Pan, "A Collaborative Cache Strategy in Satellite-Ground Integrated Network Based on Multiaccess Edge Computing," *Wireless Communications and Mobile Computing*, vol. 2021, 14 pages, 2021, Article ID 8121509. Article (CrossRef Link).
[15] N. Wang, W. Shao, S. K. Bose, and G. Shen, "MixCo: Optimal Cooperative Caching for Mobile Edge Computing in Fiber-Wireless Access Networks," in *Proc. of 2018 Optical Fiber Communications Conference and Exposition (OFC)*, pp. 1-3, 2018. Article (CrossRef Link).
[16] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," *IEEE Access*, vol. 5, pp. 6757-6779, 2017. Article (CrossRef Link).
[17] F. S. Kurniawan, L. V. Yovita, T. A. Wibowo, "Modified-LRU Algorithm for Caching on Named Data Network," in *Proc. of 2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 438-443, 2019. Article (CrossRef Link).
[18] P. Aimtongkham, C. So-In, S. Sanguanpong, "A novel web caching scheme using hybrid least frequently used and support vector machine," in *Proc. of 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1-6, 2016. Article (CrossRef Link)
[19] T. Camp, J. Boleng, V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing*, 2(5), 483-502, 2002. Article (CrossRef Link).
[20] K. Poularakis, L. Tassiulas, "Exploiting user mobility for wireless content delivery," in *Proc. of 2013 IEEE International Symposium on Information Theory*, pp. 1017-1021, 2013. Article (CrossRef Link).
[21] S. Shin, U. Lee, F. Dressler, and H. Yoon, "Analysis of Cell Sojourn Time in Heterogeneous Networks with Small Cells," *IEEE Communications Letters*, vol. 20, no. 4, pp. 788-791, April 2016. Article (CrossRef Link).
[22] X. Zhou, Z. Zhao, R. Li, Y. Zhou, J. Palicot, and H. Zhang, "Human Mobility Patterns in Cellular Networks," *IEEE Communications Letters*, vol. 17, no. 10, pp. 1877-1880, October 2013. Article (CrossRef Link).
[23] B. Banerjee, C. Tellambura, "Study of Mobility in Cache-Enabled Wireless Heterogeneous Networks," in *Proc. of 2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6, 2017. Article (CrossRef Link).
[24] B. Li, H. Zhang, and H. Lu, "User mobility prediction based on Lagrange's interpolation in ultra-dense networks," in *Proc. of 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1-6, 2016. Article (CrossRef Link).

[25] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch and G. Caire, "Femto-Caching: Wireless video content delivery through distributed caching helpers," in *Proc. of 2012 Proceedings IEEE INFOCOM*, pp. 1107-1115, 2012. Article (CrossRef Link).
[26] H. Ahlehagh and S. Dey, "Video-Aware Scheduling and Caching in the Radio Access Network," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1444-1462, Oct. 2014. Article (CrossRef Link).
[27] C. Yi, S. Huang, and J. Cai, "An Incentive Mechanism Integrating Joint Power, Channel and Link Management for Social-Aware D2D Content Sharing and Proactive Caching," *IEEE Transactions on Mobile Computing*, vol. 17, no. 4, pp. 789-802, 1 April 2018. Article (CrossRef Link).
[28] M. Chen, W. Liu, T. Wang, S. Zhang, and A. Liu, "A game-based deep reinforcement learning approach for energy-efficient computation in MEC systems," *Know-Based Syst.*, Vol. 235, Jan 2022. Article (CrossRef Link).
[29] X. B. Zhou, H. J. Ma, J. G. Gu, H. L. Chen, and W. Deng., "Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism," *Engineering Applications of Artificial Intelligence*, vol 114, 2022. Article (CrossRef Link).
[30] H. M. Zhao, J. Liu, H. Y. Chen, et al, "Intelligent Diagnosis Using Continuous Wavelet Transform and Gauss Convolutional Deep Belief Network," *IEEE Transactions on Reliability*, pp. 1-11, 2022. Article (CrossRef Link).
[31] M. Chen, W. Liu, T. Wang, S. Zhang, and A. Liu, "Deep reinforcement learning for computation offloading in mobile edge computing environment," *Computer Communications*, 175, 1-12, 2021. Article (CrossRef Link).
[32] M. Hu, J. Luo, Y. Wang, and B. Veeravalli, "Practical Resource Provisioning and Caching with Dynamic Resilience for Cloud-Based Content Distribution Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2169-2179, Aug. 2014. Article (CrossRef Link).
[33] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen and W. Zhu, "Understanding Performance of Edge Content Caching for Mobile Video Streaming," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1076-1089, May 2017. Article (CrossRef Link).
[34] W. Liu, Y. Jiang, S. Xu, G. Cao, W. Du, and Y. Cheng, "Mobility-Aware Video Prefetch Caching and Replacement Strategies in Mobile-Edge Computing Networks," in *Proc. of 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 687-694, 2018. Article (CrossRef Link).
[35] D. Niyato, D. I. Kim, P. Wang, and L. Song, "A novel caching mechanism for Internet of Things (IoT) sensing service with energy harvesting," in *Proc. of 2016 IEEE International Conference on Communications (ICC)*, pp. 1-6, 2016. Article (CrossRef Link).
[36] T. X. Tran and D. Pompili, "Octopus: A Cooperative Hierarchical Caching Strategy for Cloud Radio Access Networks," in *Proc. of 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 154-162, 2016. Article (CrossRef Link).
[37] T. X. Tran, P. Pandey, A. Hajisami and D. Pompili, "Collaborative multi-bitrate video caching and processing in Mobile-Edge Computing networks," in *Proc. of 2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pp. 165-172, 2017. Article (CrossRef Link).
[38] A. Ndikumana, S. Ullah, T. LeAnh, N. H. Tran and C. S. Hong, "Collaborative cache allocation and computation offloading in mobile edge computing," in *Proc. of 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 366-369, 2017. Article (CrossRef Link).
[39] J. George and S. Sebastian, "Cooperative caching strategy for video streaming in mobile networks," in *Proc. of 2016 International Conference on Emerging Technological Trends (ICETT)*, pp. 1-7, 2016. Article (CrossRef Link).
[40] G. Paschos, E. Bastug, I. Land, G. Caire and M. Debbah, "Wireless caching: technical misconceptions and business barriers," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 16-22, August 2016. Article (CrossRef Link).

[41] L. Breslau, Pei Cao, Li Fan, G. Phillips, and S. Shenker, "Web caching and *Zipf*-like distributions: evidence and implications," in *Proc. of IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now*, vol. 1, pp. 126-134, 1999. Article (CrossRef Link).
[42] Y. Chen et al., "Electric customer credit-rating based on entropy and Newton's law of cooling," in *Proc. of 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 2070-2074, 2017. Article (CrossRef Link).
[43] P. Karine, S. Gwendal, "YouTube live and Twitch: a tour of user-generated live streaming systems," in *Proc. of 2015 the 6th ACM Multimedia Systems Conference. Association for Computing Machinery*, New York, NY, USA, 225–230, 2015. Article (CrossRef Link).

Pingshan Liu: He is a professor at Guilin University of Electronic Technology, China. He received the PhD degree in 2014 from Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. His current research interests include Multi-access Edge Computing, Content Delivery Network, and Machine Learning.

Shaoxing Liu: He received the M.S. degree in 2022 from Guilin University of Electronic Technology, China. His current research interests include Content Delivery Network and Multi-access Edge Computing.

Zhangjing Cai: She is currently pursuing a master degree at the school of business in Guilin University of Electronic Technology, Guilin, China. Her current research interests include Multi-access Edge Computing and Content Delivery Network.

Dianjie Lu: He is a professor at School of Information Science and Engineering, Shandong Normal University, China. He received the PhD degree in 2012 from Institute of Computing Technology (ICT), Chinese Academy of Sciences. He was a Research Associate at City University of Hong Kong (2019-2020). His current research interests include IoT, smart city and cognitive wireless networks.

Guimin Huang: He is a full professor at Guilin University of Electronic Technology in China. He has published more than eighty academic papers on international journal and international conference, awarded thirteen patents of invention. His research interests include natural language processing and networks.