Hindawi
Mobile Information Systems
Volume 2021, Article ID 6625629, 14 pages
https://doi.org/10.1155/2021/6625629

Research Article

Social-Aware Edge Caching Strategy of Video Resources in 5G Ultra-Dense Network

Shijie Jia[1], **Zhen Zhou**[1], **WeiLing Li**[1], **Youzhong Ma**[1], **Ruiling Zhang**[1] and **Tianyin Wang**[2,3]

[1]*Academy of Information Technology, Luoyang Normal University, Luoyang 471934, China*
[2]*Academy of Mathematical Science, Luoyang Normal University, Luoyang 471934, China*
[3]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China*

Correspondence should be addressed to Tianyin Wang; wangtianyin79@163.com

Received 13 October 2020; Revised 5 December 2020; Accepted 21 March 2021; Published 2 April 2021

Academic Editor: Alessandro Bazzi

The video traffic offloading in edge networks is an effective method for remission of congestion of backward paths in 5G networks by continual optimization of video distribution to promote scale and efficiency of video delivery in edge networks (e.g., D2D-based near-end sharing). Because the video resources are dispersedly cached in local buffer of mobile devices of video users, the management of local video resources of video users in edge networks (e.g., caching and removing of local videos) causes dynamic variation of video distribution in networks. The real-time adjustment of local resources of users in terms of the influence levels (e.g., promotion and recession) of video sharing performance is significant for the continual distribution optimization. In this paper, we propose a novel Social-aware Edge Caching Strategy of Video Resources in 5G Ultra-Dense Network (SECS). SECS designs an estimation method of interest domain of users, which employs the Spectral Clustering to generate initial video clusters and makes use of the Fuzzy C-Means (FCM) to refine the initial video clusters. A user clustering method is proposed, which enables the users with common and similar interests to be clustered into the same groups by estimating similarity levels of interest domain between users. SECS designs a performance-aware video caching strategy, which enables the users intelligently implement management (caching and removing) of local video resources in terms of influence for the intragroup sharing performance. Extensive tests show how SECS achieves much better performance results in comparison with the state-of-the-art solutions.

1. Introduction

The video services (e.g., video-on-demand and living video streaming) provide rich viewing content for video users by making use of mobile smart devices to ubiquitous access to the Internet [1–5]. The smooth and high-definition watching quality enables the video users obtain great experience, which requires higher bandwidth and lower delay to support video data delivery from video providers to video requesters [6–8]. The new generation of communication technology 5G relies on bandwidth expansion and transmission acceleration to provide support in capacity and velocity for smooth and high-definition experience [9–11]. Moreover, the 5G makes use of the ultra-dense deployment to promote network coverage and access capability, which supports more video users to ubiquitously fetch video content via the 5G networks.

The excellent experience and convenient access of video services not only attract the large-scale video users but also speed up the increase of user scale. Obviously, the fast expansion and desired high-quality experience of user scale trigger huge bandwidth consumption, so that the unbalance between supply and demand of bandwidth results in the severe network congestion. In particular, the backhaul paths in the 5G networks inevitably are subjected by the network congestion because of dense access. The high startup delay and unbearable packet loss caused by the network congestion lead to unsmooth playback and video picture distortion, which brings severe negative influence for user quality of experience (QoE). Therefore, offloading video traffic in edge networks without intervention of 5G nodes is significant for relieving congestion levels of backhaul networks.

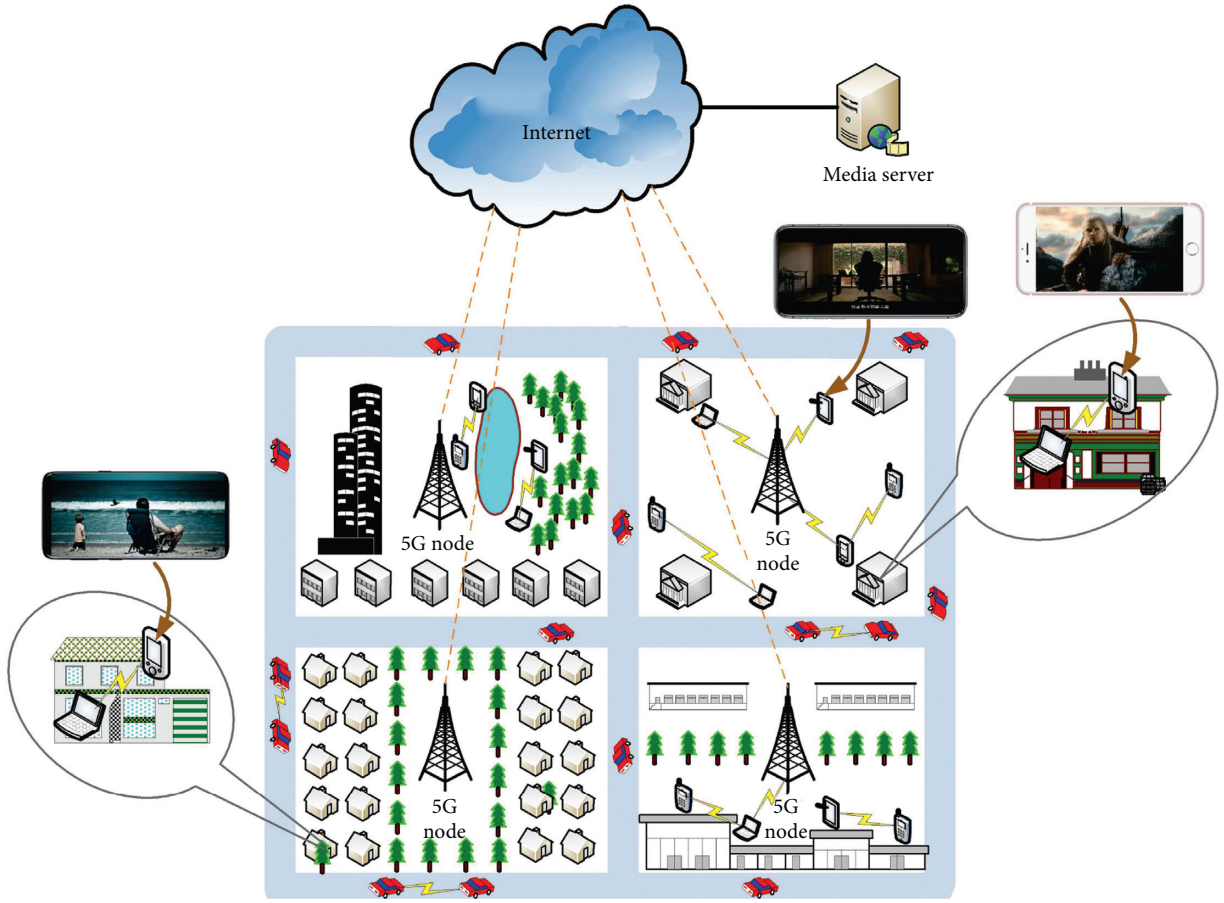

Figure 1: Video streaming services in 5G networks.

D2D communications enable mobile devices implement data transmission without intervention of 5G nodes, which become the important method for traffic offloading in underlaying networks [12–14]. As Figure 1 shows, the video users not only make use of the 5G networks accessing to the Internet to watch video content but also employ the D2D communications to fetch the video resources from other mobile devices carrying the corresponding videos. The successful traffic offloading based on the D2D communications relies on the two perspectives: (1) search of D2D matching objects which cache video resources requested by the video request nodes; (2) delivery of video data using D2D communications between request and supply nodes of video resources. The successful search of D2D matching objects which cache video resources is the important precondition for the successful delivery of video data using D2D communications. On the other hand, the fast search of D2D matching objects can efficiently reduce startup delay of video request nodes, which becomes the key factor for ensuring high QoE in terms of delay-sensitive property of video services. The successful and fast search of video providers in the range of D2D communications is vital for the offloading video traffic in the 5G networks.

The video resources are carried by the mobile devices and are distributed in networks. The users cache and remove video resources in terms of the interests, so that the dynamic distribution of videos in networks brings severely negative influence for the success ratio and time of searching video providers. On the other hand, the video resources move with the movement of mobile devices, which are dynamically distributed in geographic area of the edge networks. Therefore, the geographical movement and local replacement of video resources are main reasons of dynamic change of video resource distribution: (1) the replacement based on the self-interest change results in the increase in the risk of video search failure (the search failure increases the startup delay of video request users); (2) the geographical movement of video resources results in the decrease in the probability of D2D pairing success with one-hop neighbor relationship (the D2D pairing failure means that the video request users only make use of the multi-hops transmission to fetch video data instead of D2D communications with one-hop transmission). The numerous researchers always focus on making use of resource distribution optimization based on caching management to promote efficiency of resource sharing. For instance, Mehrabi Liu et al. propose a joint QoE-traffic optimization with collaborative edge caching by investigation of impact of collaborative mobile edge caching for both QoE and backhaul data traffic [15], which implements self-tuned bitrate selection to make the decision efficient cache replacement strategy. Thar et al. propose a deep learning-based prediction scheme, which achieves intelligent

management of resource leasing and caching [16], which can make the prediction for resource leasing and caching using the Keras and Tensorflow libraries. Jiang et al. propose a cooperative content caching policy based on multiagent reinforcement learning for the architecture without preference and historical content demands of users by designing a cooperative content caching scheme to solve the cooperative content caching problem [17]. However, the most of existing methods neglect the association relationship between interests and behaviors (requesting and caching) of edge users, which results in severe turbulence of resource distribution by the arbitrary replacement of local cached resources. The sharing performance suffers severe negative influence and the traffic load of 5G network cannot be effectively relieved. Therefore, the real-time adjustment of video distribution in terms of variation of video sharing performance is very significant for effect of D2D-based traffic offloading and user QoE.

In this paper, we propose a novel Social-aware Edge Caching Strategy of Video Resources in 5G Ultra-Dense Network (SECS). SECS designs an estimation method of interest domain of users: (1) SECS employs the Spectral Clustering method to generate initial video clusters in terms of similarity between videos by investigating video content and access behaviors of users; (2) SECS makes use of the Fuzzy C-Means (FCM) to refine the video clusters by redefining the distance between videos. The refined video clusters can be considered as the interest domain of users. A group construction method of clustering users is proposed, which relies on estimation results of similarity levels of interest domain to aggregate users, which enables users with common and similar interests to be clustered the same groups. A performance-aware video caching strategy is proposed, which implements intelligent management of local cached videos by estimating influence for the sharing performance in user groups from the three perspectives: promotion of video sharing scale, reduction of response delay, and motivation of near-end sharing. Extensive tests show how SECS achieves better results in comparison with other state-of-the-art solutions in terms of caching hit ratio, caching cost, response delay, and control overhead.

2. Related Work

The numerous researchers have focused on content edge caching methods in recent years. Qiao et al. propose a joint optimization method for the content placement and content delivery in the vehicular edge networks by making use of the trilateral cooperation communications among macrocell station, roadside units, and smart vehicles [18]. The joint optimization problem is formulated as a double time-scale Markov decision process. On the large time-scale, the content placement relies on content popularity, vehicle driving paths, and resource availability; on the small time-scale, a scheme based on vehicle scheduling and bandwidth allocation is designed, which decreases the content delivery delay. A deep deterministic policy gradient framework is proposed, which obtains a suboptimal solution corresponding to the long-term mixed integer linear programming problem. However, the formulated optimization problem relies on the precondition that the time-scale of content timeliness changes less frequently during the content delivery process. Kwak et al. propose a hybrid content caching method without the knowledge of content popularity [19]. The content caching location algorithm is designed, which supports average requested content data rates following finite service latency. By employing the Lyapunov optimization approach, a caching control problem with tight coupling between CU caching and BS caching is formulated and solved. By employing the submodularity property of the sum-weight objective function, the practical and heuristic CU/BS caching algorithms are proposed, which deals with a general caching scenario. Liang et al. focus on solving utility-oriented service entity caching problem in edge networks [20]. The positive impact brought by clients from caching a service entity in an edge server is defined as the utility in terms of variation in changed specific scenarios. A utility-based service entity caching problem is formulated, which is extended to other problems via redefinition of utility. The utility problem is proved as a NP-complete problem and is solved by a designed approximation algorithm. Zhang et al. propose an edge caching framework in integrated content-centric mobile 5G networks, which makes use of content-centric networking to achieve efficient management of content-oriented information and promotion of content delivery efficiency [21]. The content caching architecture is consisted of function entities, protocol stack, content retrieval, and edge caching approaches; the edge caching performance also is demonstrated by the authors. Saputra et al. make use of the deep learning to design proactive cooperative caching approaches, which can predict content demand of users in a mobile edge caching network [22]. The content server in edge network is responsible for collecting information of mobile edge nodes and performs the deep learning algorithm to predict content demand of whole network. A framework based on the distributed deep learning is proposed, which allows the mobile edge nodes collaborate and exchange information. The framework can decrease prediction error for content demand and does not need revealing private information of users.

Zhang et al. propose a caching placement method with the multi-winner auction in D2D-enabled caching cellular networks by investigating edge caching incentive and content caching redundancy [23]. A multiwinner edge caching auction is modeled; an optimization problem of content caching revenue maximization is formulated. A semidefinite programming is designed, which obtains an approximate optimal caching placement to reduce the content caching redundancy in a UT movement scenario. A payment strategy with Nash bargaining game based on personal profit fairness is designed and a multiwinner repeated auction based caching placement algorithm is proposed, which reduces the complexity with tiny performance loss. Zhang et al. propose a cooperative caching strategy by construction of a two-tier heterogeneous network consisted of edge servers and caching helpers, which promotes utilization ratio of storage space and caching hit probability

[24]. By making use of stochastic geometry and optimizing the caching probabilities of contents, a cooperative caching strategy is designed, which can maximize the hit probability and promotes the utilization ratio of local storage space. Zhang et al. propose a delay-optimal cooperative edge caching in wireless networks, which can optimize content placement and cluster size according to stochastic information of comprehensive measurement of network topology, traffic distribution, channel quality and file popularity [25]. The authors proposes a greedy content placement algorithm using bandwidth allocation optimization and a condition constraining the maximal cluster size based on tradeoff between caching diversity and spectrum efficiency. Liu et al. investigate heterogeneous context without preferences of user to implement management of content caching [26]. An online Bayesian clustering caching algorithm is designed, which maintains sustainable scalability by learning interactive cache hit data between users. The latent number of user groups is explained by a Dirichlet multinomial mixture model based on Bayesian generative framework (users with the same preference). Then, a dynamic clustering policy is proposed, which can obtain preferences of clusters by making use of a collapsed Gibbs sampling algorithm. A cache bandit algorithm is designed, which can support the formulation of cache decisions. Guan et al. propose an edge caching admission strategy of video content based on preference learning [27]. An information collector is designed, which collects the preference-related information without any modification of clients and video providers. A tree-structure model is designed, which learn and compress preferences of users. An explore-and-exploit method is proposed, which makes the decision of video caching. Xu et al. propose a secure edge caching scheme in mobile sensor networks [28]. A secure edge caching framework is designed, which is consisted of content provider, multiple edge caching devices, and some mobile users. An interaction model based on Stackelberg game between the content provider and edge caching devices is designed, where content provider (game-leader) determines payment strategy of secure caching service, and each edge caching device (game-follower) makes the decision for the quality of secure caching service. A zero-payment mechanism is designed, which deals with selfish behaviors of edge caching devices to stimulate content caching.

3. SECS Detailed Design

3.1. Interest Domain of Video Users. The interest for the video content is main reason to drive the video users requesting video resources. The various interests for the different video content make the video users dedicate fetching the desired video resources. However, the user interests for video content are dynamically variational. When the video content is in the range of desired video kind for the video users, the users send the video request to fetch and store the video resources; when the users have watched the video content, the user interest levels for the watched videos fast weaken, so that the watched videos may be removed in the local buffer. The variation of user interests determines the change of requested and cached videos, which brings the severe negative influence for the video distribution. This is as the video system dispatches the video resources to respond the video request of users and efficiently makes use of the buffer space (e.g., removing unpopular videos and caching popular videos) to support the large-scale access. Obviously, the interest analysis is important for understanding and predicting behaviors (e.g. request, caching and replacement of video resources) of video users, which supports beforehand or real-time adjustment of video resource distribution to balance supply and demand and promote video sharing efficiency. Table 1 lists the definition of symbols in the whole paper.

Measurement of user interests relies on the videos which have been watched by the users. However, the single video does not reflect the real interest of users. A watched video only denotes the user has interest for the video, which does not help the video system understand the intention that the user watches the video and predicts the video requested by the user in the future. If the watched videos are aggregated, the clustered video sets not only denote interest kind for the video content but also describe the boundary of interested video resources. Further, the interest domain can be used to estimate the request probability of video users for any video, predict the requested video resources in the future, and cluster the video users with the same or similar interest. The traditional video clustering methods make use of the structured video information to measure the similarity values between videos. For instance, a video v_i can be denoted as $v_i = (a_1, a_2, \ldots, a_k)$, where a is any attribute of v_i (e.g., title, actor and abstract of video). If $v_i = (a_1, a_2, \ldots, a_k)$ can be considered a vector, the similarity value between v_i and v_j can be defined as the angle cosine between two vectors v_i and v_j. However, the measurement of similarity between videos only investigates the relation levels between video content and does not involve the influence factors from the request behaviors of video users. For instance, the continuous two videos in a TV play series may have the similar actor, but the plot may be different for the two videos. The single measurement based on the content similarity cannot find the two videos belong to the same TV play series. The relation from the request behaviors of video users should be considered as the measurement parameter for estimation of video similarity. Therefore, the merged similarity based on content and request relation can be defined as

$$S_{i,j} = sc_{i,j} \times sa_{i,j}, \quad sc_{i,j} \in [0, 1], i \neq j, \tag{1}$$

where $sc_{i,j}$ is the content similarity between v_i and v_j; $sa_{i,j}$ is the similarity of user access relation between v_i and v_j. In terms of the abovementioned traditional measurement method of video similarity, v_i and v_j can be the two vectors which include the same attributes, respectively. The angle cosine between the two vectors corresponding to v_i and v_j can be considered as the content similarity $sc_{i,j}$ between v_i and v_j. The measurement method of value of $sa_{i,j}$ needs to investigate the association relationship between v_i and v_j in the request behaviors of video users. Let $\log = (l_1, l_2, \ldots, l_n)$ denote the all logs of video request where any item l_i in log is

Table 1: Notations used by the paper.

Parameters	Definition
a	Attribute of video
sa	Similarity of access relation between videos
f	Number of request association relation
sv_i	Vector of video similarity values
d_i	degree value of any video v_i
λ	Characteristic value of L
$\mathbb{U}$	$n \times k$ matrix Of characteristic vectors
CV	Set of video clusters
$M_{i,j}$	Membership of v_i belonging to c_j
$I_j(v_i)$	Interest level of u_j for v_i
$\overline{T}_k^{(d)}$	Average wait delay
l_i	Viewing log of user u_i
$P_{j,h}^{(p)}$	Probability of pushing video with encounter
$f_{j,h}$	Frequency of video sharing between users
$\overline{T}_k^{(d)}$	Average wait delay
$P_{j,h}^{(r)}$	Probability of requesting video with encounter
λ_i	Rate of request and handling
$\delta_{j,h}$	Acceptation probability
sc	Content similarity between videos
S	Video similarity
n	Total number of video logs
$\mathbb{R}$	$m \times m$ matrix Of video vectors
L	Standardized laplacian matrix
$\mathbb{C}$	Set of characteristic values of L
c	Video cluster
$d_{i,j}$	Distance between v_i and v_j
$m_{i,k}$	Membership of video to cluster centric
$P_{j,h}(v_i)$	Acceptance probability of u_j for v_i
ID_i	Interest domain of user u_i
$I_j(v_i)$	Interest level of u_j for v_i
$\overline{T}_k^{(s)}$	Average serving capacity of user
g	Number of request messages
$P_{j,h}^{(e)}$	Probability of users becoming neighbors
$P_{j,h}^{(p)}$	Probability of pushing video with encounter
μ_i	Rate of request and handling
P (r)	Probability of r-th requested video

the video access log of a user i in the video system. If a log l_i includes v_i and v_j and the location of v_i and v_j in l_i is adjacent, v_i and v_j have a request association relation in l_i. The value of $sa_{i,j}$ can be defined as

$$sa_{i,j} = \frac{f_{i,j}}{n}, \quad sa_{i,j} \in [0,1], \tag{2}$$

where $f_{i,j}$ is the number of request association relation of v_i and v_j and n is the total number of all logs in log. If the number of request association relation of v_i and v_j is 0, $f_{i,j} = 0$ and $sa_{i,j} = 0$. The similarity values among v_i and all videos form a vector $sv_i = (S_{i,1}, S_{i,2}, \ldots, S_{i,m})$ according to equation (1) where $S_{i,i} = 0$ and m is the total of all videos. All vectors corresponding to the videos can form an $m \times m$ matrix $\mathbb{R}$. Because $S_{i,j} = 0$ and $i = j$, the values of similarity in the diagonal line of $\mathbb{R}$ are 0. We employ the Spectral Clustering method to cluster videos in terms of the matrix $\mathbb{R}$ of video similarity [29–31]. Because the Spectral Clustering method is well known, we briefly introduce the process of clustering video. The degree value of any video v_i can be defined as the total sum of similarity values among v_i and other videos, as follows:

$$d_i = \sum_{j=1}^{m} S_{i,j}, \tag{3}$$

where m is the total number of all videos. In fact, d_i is the sum of all items in i^{th} line of $\mathbb{R}$. The degree matrix of similar matrix $\mathbb{R}$ can be built and be denoted as D. The standardized Laplacian matrix L can be calculated according to $L = D^{-1/2}(D - \mathbb{R})D^{-1/2}$. The m characteristic values of L can be further obtained and are sorted according to ascending sequence of values of all items in $\mathbb{C}$, namely, $\mathbb{C} = (\lambda_1, \lambda_2, \ldots, \lambda_m)$. Any item λ_i in $\mathbb{C}$ is equal or greater than 0. The subset $(\lambda_1, \lambda_2, \ldots, \lambda_k)$ in $\mathbb{C}$ is extracted. The characteristic vectors $(u_1, u_2, \ldots, u_k)$ are calculated corresponding to $(\lambda_1, \lambda_2, \ldots, \lambda_k)$, where the dimension of any vector is n. The characteristic vectors $(u_1, u_2, \ldots, u_k)$ form a $n \times k$ matrix $\mathbb{U}$, where n is the number of videos and k is the purposed number of video clusters. The most of Spectral Clustering methods employ the k-means method to divide the n videos into k clusters based on the $n \times k$ matrix $\mathbb{U}$. However, the clustering exactitude level of Spectral Clustering method relies on the estimation accuracy of similarity between samples and the clustering performance of k-means method depends on the selection of center samples of clusters [32–34]. Therefore, after using the k-means method, the k video clusters can be obtained and be defined as $CV = (c_1, c_2, \ldots, c_k)$. Each item c_i in CV is denoted as $c_i(v_k) = (v_a, v_b, \ldots, v_h)$, where v_k is the centric item of c_i.

The accuracy of video clustering is very important for generation of interest domain of users and estimation of video request probability of users. If the accuracy of video clustering is low, the video clusters include videos which are dissimilar with each other, so that the interests of users are not definitely obtained and the probabilities of video request of users are not precisely estimated. This leads to guideless caching and replacement of video resources bring severely negative influence for video sharing performance and user QoE. Therefore, the video clusters based on the Spectral Clustering are considered as the initial clusters and should be refined to promote the accuracy of video clusters (e.g., high cohesion between videos and low coupling between clusters). We employ the Fuzzy C-Means (FCM) method to refine the initial video clusters [35]. The objective function of FCM can be defined as

$$\begin{aligned} J &= \sum_{i=1}^{|V|} \sum_{j=1}^{|CV|} M_{i,j} d_{i,j}, \\ \sum_{j=1}^{|c_i|} M_{i,j} &= 1, \end{aligned} \tag{4}$$

where $d_{i,j}$ is the distance between two videos v_i and v_j and $d_{i,j} = S_{i,j}$; $|V|$ is the total number of video resources and $|CV|$

returns the total of all items in video cluster set CV; and $M_{i,j}$ is the membership of v_i belonging to c_j and can be defined as

$$M_{i,j} = \sum_{c=1}^{|CV|} \left(\frac{m_{i,k}}{m_{i,c}} \right)^{2/m-1}, \quad (5)$$

where $m_{i,k}$ is the membership between v_i and centric item of c_k and can be defined as

$$m_{i,k} = S_{i,k} \times w_{i,j},$$
$$w_{i,j} = \frac{f_{i,k}}{|c_k|}, \quad (6)$$

where $S_{i,k}$ is the content similarity between v_i and centric item v_k of c_k, $w_{i,j}$ is a weight value, and $f_{i,k}$ is the number of c_k's items which have the co-occurrence relationship with v_i. For instance, if v_i and any item v_j in c_k jointly are included in any log, v_i and v_j have the co-occurrence relationship. In fact, $f_{i,k}$ is the number of items in c_k which jointly are included with v_i in logs. Obviously, $f_{i,k} \leq |c_k|$ and $w_{i,j} \in [0, 1]$. sa reflects the continuity between videos for the request behaviors of users; w investigates range that a video has the association relationship with the items in a video cluster for the requested content of users. Because the FCM makes use of recalculating membership values of between each item and all clusters to promote intracluster cohesion and reduce intercluster coupling after adjustment of centric item of cluster, the FCM has multiple refinement round and intraround recalculation. In order to clearly describe the refinement process based on the FCM, J_g^h denotes value of objective function in equation (4), where g is the number of refinement round and h is the number of recalculation in the current round. The following process shows refinement of clusters in CV:

Step 1: checks whether all items in CV are marked or not. If all items in CV have been marked as "refined", implements Step 7; Otherwise, if all items in CV are not marked, extract a cluster c which is not refined in the current refinement round from CV.

Step 2: the objective function value of h^{th} calculation of g^{th} round is defined as J_g^h. The item which has the minimum distance with centric node of c is removed from S.

Step 3: the centric node of S should be reselected because of removing an item in c. After reselection of centric item of c, the value J of objective function needs to be recalculated and is marked as J_g^{h+1}.

Step 4: if $J_g^{h+1} > J_g^h$, the removed item from c can be considered as a noise item and is added into a noise set SR; c is marked and the mark status is "refined"; otherwise, if $J_g^{h+1} \leq J_g^h$, the removed item from S cannot be a noise item and is re-added into c; The original centric item of c still is centric item of c; c is marked and the mark status is "unrefined".

Step 5: If CV still includes "unmarked" items, returns Step 1; otherwise, implements Step 6.

Step 6: if all items in CV are marked as "refined" or "unrefined" and the number of "refined" items is equal or greater than 1, the items in CV still need to be refined. The number of refinement round is $g + 1$. All items in CV are redefined as "unmarked" and returns to Step 1; otherwise, if all items in CV are marked and the number of "refined" items is 0, implements Step 7.

Step 7: the current refinement iteration is terminated, and all items in CV are considered as "refined."

Because there is only one nested iteration process of membership calculation based on matching video similarity in the above refinement process, the complexity of the above refinement process is $O(n^2)$. The initial video clusters in CV are refined in terms of the above refinement process, and the set consisted of refined clusters still is defined as CV. If the viewing log l_i of a video user u_i is defined as $l_i = (v_a, v_b, \ldots, v_k)$ and the items in l_i, respectively, belong to video clusters $c_a, c_b, \ldots, c_k$, the interest domain of u_i is considered as $ID_i = (c_a, c_b, \ldots, c_k)$. When a new video v_k is added into V, v_k is considered as a member in the cluster c_i where the centric item of c_i has the largest similarity value with v_k among all clusters. In order to avoid the negative influence for the accuracy of clustering videos from the new videos, after v_k is added into c_i, the centric item of c_i should be reselected. $\overline{m}$ and $\overline{m}\prime$ can be calculated, where $\overline{m}$ and $\overline{m}\prime$ are the average membership between all items and centric item before and after v_k joining into c_i, respectively. If $\overline{m}\prime > \overline{m}$, v_k is added into c_i and the reselected centric item becomes the new centric item of c_i; otherwise, v_k forms a new cluster and the original centric item still act as the centric item of c_i.

3.2. User Group with Common Interest Domain. The interest domain of video users not only denotes range of requested videos but also can be used to measure and predict video sharing among users. For instance, if the two users have the same interest domain, they may supply desired videos with each other by pull and push; otherwise, if the two users have the different interest domain, the video providers cannot supply the requested videos for the video requesters and the video requesters cannot receive the pushed videos from the video providers. Video sharing between users with common interest range not only promotes QoS of video system and QoE of users but also increases utilization of cached resources and energy-efficiency levels of mobile devices. Classifying users with common interests into the same groups is very important for the promotion of video sharing performance.

Let $US = (us_1, us_2, \ldots, us_m)$. Each user has the definite interest domain and the interest domain directly shows content and range of representational preference of users. The set consisted of centric items corresponding to interest domain of users can be represented as the interest domain of users. The users which have the same or similar interest domain can be allocated into the same user groups, as follows.

Step 1: if $US = \varnothing$, implements Step 5; otherwise, if US includes the items which have the same interest domain, implements Step 2; otherwise, if the items in US do not have the same interest domain, implements Step 3.

Step 2: extracting an item subset uss from US where the uss includes the most items among all the item subsets with the same interest domain. uss can be considered as a user category and an item is selected as centric item of uss. $US = US - uss$, where $US - uss$ denotes the difference set between US and uss, which means that the classified items are removed from US. uss is added into the set USS; return Step 1.

Step 3: if $USS = \varnothing$ (all items in US do not have the same interest domain), implement Step 4; otherwise, an item us in US is selected and the similarity values of interest domain of us and all subsets in uss are calculated according to the equation $IRS = |ID_i \cap ID_j|/|ID_i \cup ID_j|$. ID_i and ID_j are the interest domain of us_i and uss_j, respectively; $|ID_i \cap ID_j|$ and $|ID_i \cup ID_j|$ return the number of intersection and union of interest domain of us_i and uss_j, respectively. us joins into the subset which has the largest similarity value with us among all subsets in USS. us is removed from US and returns Step 1.

Step 4: each item in US is considered as a subset, is added into USS, and is removed from US; returns Step 1.

Step 5: the current iteration of user classification is terminated.

The above aggregation process can be described in Algorithm 1. Because there are the one nested iteration process of similarity calculation based on matching interest domain in Algorithm 1, the complexity of Algorithm 1 is $O(n^2)$. The users with common interest domain can be aggregated in terms of the above process and form a user group set $USS = (uss_1, uss_2, \ldots, uss_m)$. The users in each item of USS have the same or similar interests for video content. The union set of all users of each item in USS is considered as the interest domain of current user group.

4. Video Caching Management Based on Sharing Performance Awareness

The current mobile devices in edge networks have relatively high performance (e.g., fast computation and large storage). However, the development of video quality (e.g., blu-ray video) brings the fast increase of video size, so that the storage capacities of mobile devices become the relatively finitude. When the local videos occupy the large number of storage resources, the users have to remove some local videos in order to store new videos in the future. However, the caching and removing of local videos bring the immeasurable influence for the video resource distribution in the edge networks. For instance, when a user stores a video into local buffer, the supply capacity corresponding to the cached video is enhanced in the edge networks. The user not only makes use of the cached video to promote the supply capacity of upload bandwidth in the same user group, but also provides video data for users in edge networks. The sufficient supply of video resources can reduce the queued response delay of video request and promote the probability of near-end video sharing (e.g. D2D communications). However, when the popularity of a video decreases, the superfluous supply also wastes the local storage resources of mobile devices. On the other hand, when a user removes a video from local buffer, the video supply decreases regardless of user group or edge networks. The lacking supply of videos can increase the queued response delay of request and reduce the probability of near-end video sharing. However, when the popularity of a video decreases, the removing for the superfluous video caching can promote the use efficiency of local buffer. Therefore, the influence for the video sharing performance should be estimated before the caching or removing of local videos. We investigate the video sharing performance in terms of the three perspectives of promotion of video sharing scale, reduction of response delay and motivation of near-end sharing.

4.1. Promotion of Video Sharing Scale. The videos rely on visible content to attract accessing of users. If the videos have irresistible content, the positive request of users promotes popularity of videos and diffusion of video copies in edge networks. After the information of videos (e.g., title and abstract of videos) is obtained by the users, the users which are interested in the videos want to fetch the video resources by sending request messages; instead of the positive request, when the users may receive the pushed messages of video information from other users, they make the decision of receiving the pushed videos.

The interest levels are main driving factors of users requesting video resources. When a video v_i starts to disseminate in networks, the interest level of a user u_j for v_i can be defined as

$$I_j(v_i) = S_{i,k}\frac{N_k}{N_j}, \tag{7}$$

where $S_{i,k}$ denotes the similarity value between v_i; v_k and v_k is the centric item of video cluster c_k; N_k is the number of videos which belongs to c_k and has been watched by u_j; N_j is the total number of videos which have been watched by u_j; and N_k/N_j denotes the interest level of u_j for the interest subdomain c_k. The higher the value of N_k/N_j is, the stronger the intentions of u_j for requesting videos in c_k is. If $S_{i,k}$ is high, the membership level between v_i and c_k is strong; If $S_{i,k}$ is low, v_i has the weak similarity relationship with the most of items in c_k. In fact, $S_{i,k}$ can be considered as a weight of N_k/N_j. The larger the value of $I_j(v_i)$ is, the higher the probability of u_j requesting v_i is. On the other hand, when u_j receives a push message about v_i, u_j not only investigates the interest level for v_i but also considers the social relationship. The acceptance probability of u_j for a pushed v_i can be defined as

$$P_j, h(v_i) = \overline{S_j}\frac{f_{j,h}^k}{f_{j,h}}, \qquad \overline{S_j} = \frac{\sum_{c=1}^{m} S_{i,c}}{m}, \tag{8}$$

where m is the number of videos which are successfully pushed by u_i in the video cluster corresponding to v_i; $\overline{S_j}$ is

```
1: /* US is user set; USS is set of user groups;
2: TS1 and TS2 are empty sets*/;
3: for (i = 0; i|US|; i++)
4:       US[i] is added into TS1;
5:       for (j = 0; j|US|; j++)
6:          if US[i] has same domain with US[j]
7:             US[j] is added into TS1;
8:          end if
9:       end for
10:    if |TS1| < 2
11:       US[i] is removed from US and is added into TS2;
12:    else items in TS1 are removed from US;
13:       TS1 is added into USS;
14:    end if
15:    TS1 is set to empty set;
16: end for
17: all items in TS2 are added into US;
18: if US is not ∅
19:    if USS is ∅
20:       for (i = 0; i|US|; i++)
21:          US[i] is a group and is added into USS;
22:       end for
23: else for (i = 0; i|US|; i++)
24:          computes similarity values between US[i] and all centric items in USS;
25:          if centric item of subset uss_k has the largest similarity with US[i]
26:             US[i] is added into uss_k;
27:          end if
28:       end for
29:    end if
30: end if
```

Algorithm 1: Aggregation Process of users.

the mean value of similarity between v_i and videos which are successfully pushed by u_i in the video cluster corresponding to v_i; $f^k_{j,h}$ is the sharing frequency of videos in c_k between u_j and u_h; and $f_{j,h}$ is the frequency of video sharing between u_j and u_h. The larger the value of $f^k_{j,h}/f_{j,h}$ is, the closer the social relationship between u_j and u_h is.

4.2. Reduction of Response Delay. The users make use of the request and push to disseminate a new video v_i. If v_i has high popularity, the large number of users may request v_i or accept the push of v_i. In the process of users requesting v_i, the supply users which have cached v_i in local buffer receive the request messages and send video data to the request users. The request users receive video data, watch the video content and store the video into local buffer. At the moment, the video copies are generated by the sharing (diffusion) between users. When the number of request users fast increases, the video system needs to make use of the upload bandwidth of users which have cached the copies in networks to provide the video resources for the request users.

Let λ_a and μ_a be the rate of request and handling, respectively. $\lambda_a = N_r/T_a$, where N_r is the number of generated request messages during the time span T_a; $\mu_a = N_h/T_a$, where N_h is the number of handled request messages during the time span T_a. $\lambda_a > \mu_a$ during the same time span T_a means that the request of users cannot be met due to the deficient supply of requested videos. Because the supply users with limited handling capacities (e.g., low bandwidth, storage, and computation) cannot fast deal with the mass request messages in terms of "early come early service" principle, a large number of request users need to wait for handling request messages. Obviously, the deficient supply leads to the long wait delay. $\lambda_a \leq \mu_a$ during the same time span T_a means that the request of users can be met due to the sufficient supply of requested videos. The supply users have low handling capacities, but the sufficient number of supply users enables the request messages be uniformly distributed to the supply users. The request messages can be handled in time, and the wait delay of request users can be reduced.

If a user $u_j \in uss_k$ has cached and watched a video v_i, u_j has lost the interest for v_i. In order to save the storage space and promote the utilization of storage resources, u_j replaces videos in local buffer. Before u_j removes videos in local buffer, u_j needs to estimate the influence for intra-uss_k supply capacity of v_i. We assume that the request messages generated by users of uss_k arriving intra-uss_k providers of v_i meet the $M/G/1$ queuing model. The serving capacity T_{u_j} of u_j can be obtained in terms of our previous work [36]. In fact, T_{u_j} denotes that the sum of time of handling request message and time of delivering video data. The average serving capacity of users in uss_k can be defined as

$$\overline{T}_k^{(s)} = \frac{\sum_{i=1}^{m} T_{u_i}}{m}, \tag{9}$$

where m is the number of users which have cached v_i in uss_k. Let g be the number of request messages generated by intra-uss_k users during a future time span t_a. The value of g can be calculated according to the following equation:

$$g = \left|uss_k\right| - N_k^l - N_k^u, \tag{10}$$

where $|uss_k|$ is the number of all users in uss_k; N_k^l is the number of users which have watched v_i and lose the interest for v_i; N_k^u is the total number of users which are uninterested in v_i and users which have the low interest $(I(v_i) < I_T)$, where I_T is the threshold value of interest; $I(v_i) \geq I_T$ denotes that users are interested in v_i; and $I(v_i) < I_T$ denotes that users are uninterested in v_i. Let h be the number of users which have cached v_i during t_a. $g \leq h$ denotes that the supply of v_i in uss_k meets the demand for request of v_i, so that the wait delay is 0; if $g > h$, the average wait delay can be defined as

$$\overline{T}_k^{(d)} = \frac{(g-h)}{h} \times \overline{T}_k^{(s)}. \tag{11}$$

When the supply of v_i in uss_k cannot meet the demand for request of v_i, u_j makes the decision of replacement of v_i according to the relationship between request and supply of v_i after the generation of new copies of v_i.

4.3. Motivation of Near-End Sharing. The delivery quality of video data relies on transmission path of video data. As we know, the transmission delay is the sum of forwarding delay of relay nodes. If the transmission path has the large number of relay nodes, the transmission delay may be high; if the number of relay nodes in transmission path is less or is 0 (the relationship between requesters and providers of videos is one-hop neighbor), the transmission delay is low or is 0. Moreover, if the transmission path has multiple relay nodes which have high mobility, the dynamic change of geographic location of relay nodes leads to the dynamic variation of transmission path. The variation of transmission path further results in the increase of packet loss and rise of forwarding delay.

In order to promote the transmission performance of video data and user quality of experience, the data transmission between one-hop neighbor nodes should be accelerated. Let g denote the number of users which have the interest for v_i in uss_k $(I(v_i) \geq I_T)$ during a future time span t_a. The probability of u_j and u_h becoming one-hop neighbors in g users during t_a is defined as

$$P_{j,h}^{(e)} = \frac{f_{j,h}}{f_j}, \tag{12}$$

where $f_{j,h}$ is encounter frequency between u_j and u_h during t_a and f_j is the total encounter frequency between u_j and other users in uss_k during t_a. If u_j has cached v_i, the probability of u_h requesting v_i to u_j during the encounter period time of u_j and u_h is defined as

$$P_{j,h}^{(r)} = P_{j,h}^{(e)} \times I_h(v_i). \tag{13}$$

The probability of u_j pushing v_i to u_h during the encounter period time of u_j and u_h is defined as

$$P_{j,h}^{(p)} = P_{j,h}^{(e)} \times P_j, h(v_i). \tag{14}$$

4.4. Video Caching Management Strategy. The video sharing in edge networks can offload the traffic to reduce the load of core networks, which reduces the risk of network congestion. By optimizing distribution of video resources, the video resources can be efficiently allocated in edge networks and promote the utilization efficiency of storage resources. The adjustment of video distribution relies on the demand of video requesters. The performance of video sharing directly reflects the game relationship between supply and demand. The management of video caching is the main tool of adjustment of video distribution. The management of cached videos in local buffer has the important influence for the user quality of experience in terms of the three perspectives of promotion of video sharing scale, reduction of response delay, and motivation of near-end sharing. The management of cached videos based on the performance awareness of video sharing is an efficient method for the effort of distribution optimization and video sharing. We design a video caching management strategy based on the performance awareness of video sharing for a user u_j in uss_k.

u_j has watched v_i and has cached v_i in local buffer. Let λ_i and μ_i be rate of request and handling in uss_k, respectively. $\lambda_i^k > \mu_i^k$ denotes that the number of users which store v_i is less than the number of requesting v_i in uss_k, so that u_j does not remove v_i in local buffer. At the moment, u_j estimates the dissemination scale of v_i in uss_k to make the decision of video pushing. u_j can be aware of the users in uss_k which are interested in v_i according to equation (10). u_j divides the g users into the two subsets: $sg_1 = (u_a, u_b, \ldots, u_p)$, where each item u_c in sg_1 has $I_c(v_i) > P_{j,c}(v_i)$ and has more willing to make an active request than accepting pushing of v_i from u_j; $sg_2 = (u_d, u_f, \ldots, u_q)$, where each item u_h in sg_2 has $P_{j,h}(v_i) > I_h(v_i)$ and has higher probabilities of accepting pushing of v_i from u_j than those of active request. u_j preferentially pushes v_i to items in sg_2, which can promote supply capacities in uss_k by pushing v_i with high acceptation probability. u_j needs to select a user in sg_2 which has high acceptation probability and low delivery delay according to the following equation:

$$\delta_{j,h} = P_{j,h}^{(p)} \times \frac{\overline{T}_k^{(d)}}{T_{j,h}}, \tag{15}$$

where $T_{j,h}$ is the predicted delivery time based on pushing from u_j to u_h. The parameters in $T_{j,h}$ (e.g., packet loss rate and transmission delay) can be obtained by sending detection messages to estimate communication quality of transmission path during a small time slot. A user u_b has the largest value of δ among all items in sg_2; a user u_c has the largest value of $\chi_{j,c} = P_{j,h}(v_i) \times (\overline{T}_k^{(d)}/T_{j,c})$ among all items

in sg_2. If $\delta_{j,h} > \chi_{j,c}$, u_j sends data of v_i to u_b in order to generate copies; otherwise, if $\delta_{j,h} \le \chi_{j,c}$, u_j sends data of v_i to u_c. After u_j has delivered v_i, u_j makes the decision of caching v_i in terms of the relationship between λ_i^k and μ_i^k.

If $\lambda_i^k \le \mu_i^k$, uss_k has enough supply capacity relative to request scale. Let g denote the number of users which are interested in v_i; let q denote the number of users in uss_k which store v_i. If $g \le q$, u_j can remove v_i in local buffer after u_j informs users which store v_i by sending messages; if $g > q$, u_j estimates the number of users whose state transition from "interested" to "requested" by the following equation:

$$\theta_k^i = \frac{\sum_{c=1}^{|ts_k|} N_c^r / N_c^i}{|ts_k|}, \tag{16}$$

where ts_k is a set of time slots; $|ts_k|$ is the number of time slots in ts_k; N_c^r and N_c^i are the number of users which request v_i and are interested in v_i during each time slot t_c in ts_k, respectively; N_c^r/N_c^i denotes the transition ratio from "interested" to "requested;" and θ_k^i denotes the average value of transition ratio from "interested" to "requested" during ts_k. If $\theta_k^i \times g > q$, u_j needs to continuously store v_i in local buffer; if $\theta_k^i \times g \le q$, u_j can remove v_i in local buffer.

5. Testing and Test Result Analysis

5.1. Testing Topology and Scenarios. We conduct numerous simulation tests to compare the performance of SECS with current widely used caching policy random caching (with probability equal to 0.5) and state-of-the-art solution OCP [37], based on NS-3. The parameter settings are given as follows: a square scenario with $6000 \times 6000\,m^2$ area is implemented, and 500 mobile nodes move in the scenario according to the random way point model. In this model, mobile node first randomly selects a point as the destination by uniform distribution and then determines a velocity and begins to move to the chosen point. After reaching this point, the node repeats the above destination and velocity selection process and begins to move again. The velocity of mobile nodes ranges from 10 m/s to 40 m/s. We reset the physical and MAC layer and modulation schemes of network units according to the 5G industrial standardization.

For evaluating the caching performance, we consider the videos as the transmission data. This is because video service is the driven force behind the current rapid growth of network traffic and the key applications of edge caching. In our simulation, 40 different videos are introduced and shared among the mobile users. The playback bitrate of the video is 2000 kbps and time length of a video is 120 s. We further divide each video into 60 small chunks which is 2 s long and with the size of 500 KB. The size of caching space at each mobile node is set 20 to 40 chunks. We further describe the distribution of request behaviors of mobile users at video-level by Zipf distribution [38]. Namely, given a video set of N videos, the probability of the r-th most popular video being requested by users can be given by [39, 40]

$$P(r) = \frac{\left(\sum_{k=1}^{N} 1/k^{\alpha}\right)^{-1}}{r^{\alpha}}, \tag{17}$$

where α is the Zipf parameter and set to 0.8. For the chunk-level request behaviors, after determining the video to watch, the user sequentially request the chunks of this video for the continuous playback. To implement the OCP caching policy, we uniformly deploy 25 base stations as the coordinators and source video suppliers in OCP.

5.2. Performance Evaluation. We compare the performance of SECS with OCP and random cache in terms of the caching hit ratio, caching cost, response delay, and control overhead, respectively.

Caching hit ratio (CHR): in edge caching, if one node receives a video request whose corresponding chunk is in local cache, then we consider it is a cache hit event; otherwise, it is a cache miss. We define the cache hit ratio as the average ratio between the number of the cache hit events and total number of issued requests (the sum of cache hit and miss events). A higher caching hit ratio indicates that more requests are satisfied by the nearby mobile nodes, namely, a better edge caching utilization and shorter transmission distance. In contrast, a lower caching hit ratio means the edge caching is not fully utilized and mobile users may still have to access the video from far end. Figures 2 and 3 show the CHR of the three solutions with the varying of simulation time, when the size of caching space is 20 and 40 chunks, respectively. We observe that SECS achieves the highest CHR among three solutions. In both figures, we observe that the SECS and OCP first experience an increasing trend and then enter a stable phase when 200 s. The random cache first reaches the highest CHR before 100 s and then slightly decreases. The reason for random cache's decreasing is large number of requests also results in a frequent cache miss and caching replacement, which significantly decrease the caching utilization. After entering the stable phase, the CHR of SECS is about 8% (5%) and (61%) higher than that of OCP and random cache when caching space is 20 (40), respectively.

OCP and SECS having better performance than random cache is mainly because these two solutions investigate the users' demand and supplies to achieve higher caching utilization; yet, random cache only simply sets a probability to cache the content which cannot maintain the balance between video demand and supply. Instead of setting a same caching time threshold for all mobile nodes as in OCP, SECS considers the sharing capacity of each mobile users to achieve the flexible management on caching space. With such design, SECS can provide more accurate cache placement and thereby a higher CHR. Besides, SECS also replace the content in cache according to the sharing capacity which further improves the cache utilization.

Caching cost (CC): we define the total number of caching events during the simulation as CC. Figures 4 and 5 show the CC of three solutions with the caching space 20 and 40, respectively. Higher CC means that the caching strategies may consume more storage or energy resource on performing the

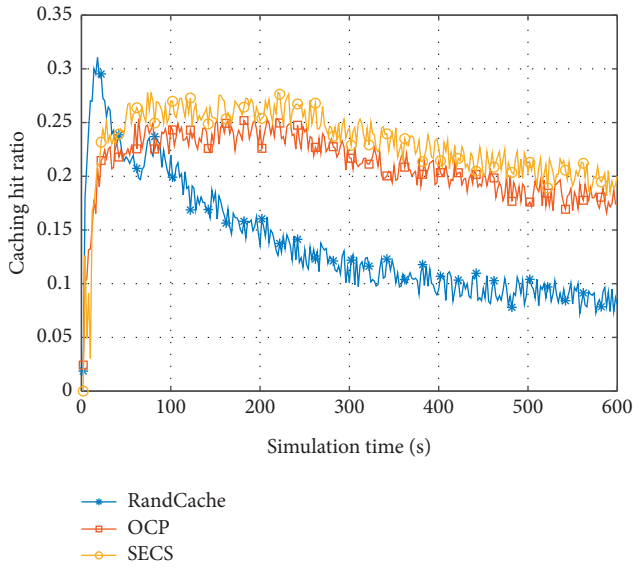

Figure 2: Caching hit ratio against simulation time when caching space is 20.

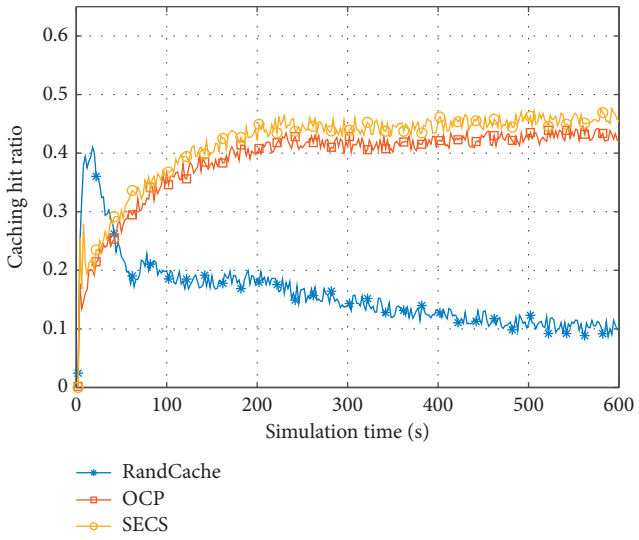

Figure 3: Caching hit ratio against simulation time when caching space is 40.

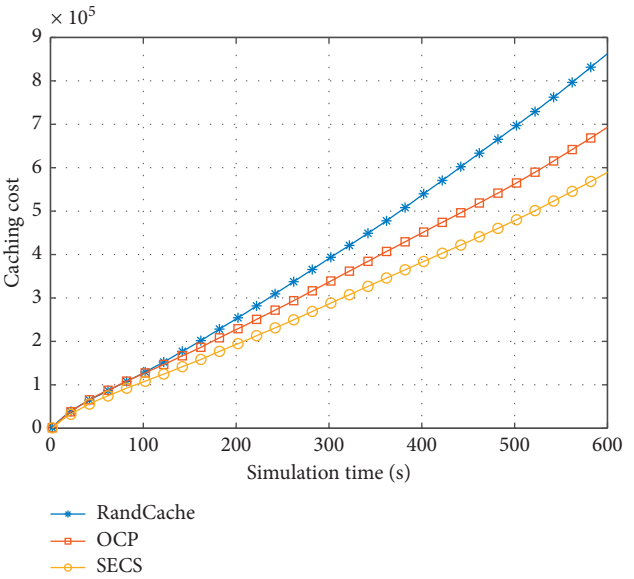

Figure 4: Caching cost against simulation time when caching space is 20.

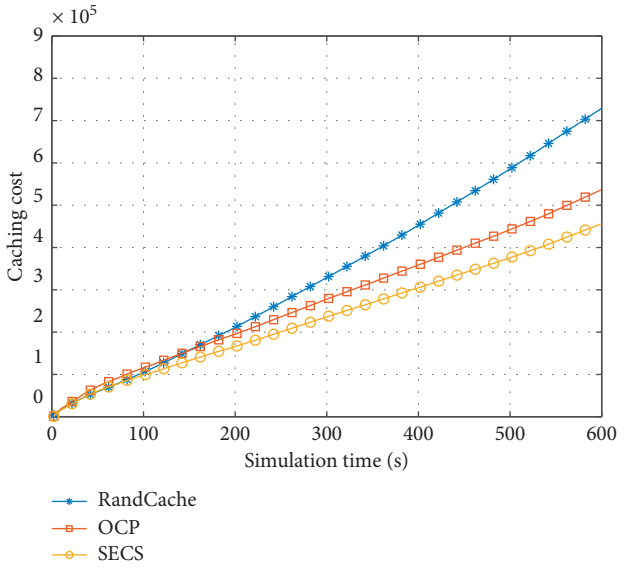

Figure 5: Caching cost against simulation time when caching space is 40.

caching operation, which may harm the system sustainability. In figures, all curves reveal a linear increase trend due to the continuous caching operations during the simulation. Comparing the two figures, we observe that all solutions with caching space 40 has lower CC when comparing with caching space 20. This is because a larger caching space can avoid the unnecessary caching replacement, which in turns reduces the caching cost. In both figures, SECS achieves the lowest CC among three solutions. For example, when caching space is 20, SECS is 8% and 17% lower than OCP and random cache when 400 s, respectively. SECS's advantage is expended when caching space reaches 40, which is 13% and 19% lower than OCP and SEC when 400 s, respectively.

SECS has the lowest CC mainly because the caching decision making based on sharing capacity provides more accurate caching placement, which avoid frequent caching eviction and replacement. Hence, SECS can significantly reduce the CC. OCP formulates the caching optimization problem by jointly considering the caching cost and system load, which reduces the unnecessary caching operations and thereby achieves lower CC than that of random cache. However, OCP overlooks the caching replacement and simple uses the LRU caching replacement policy, which yields to a higher CC than SECS.

Response delay (RD): we define the response delay as the latency between mobile users sending out request and receiving the first packet of requested content. Low RD indicates that mobile users can access video content from nearby users

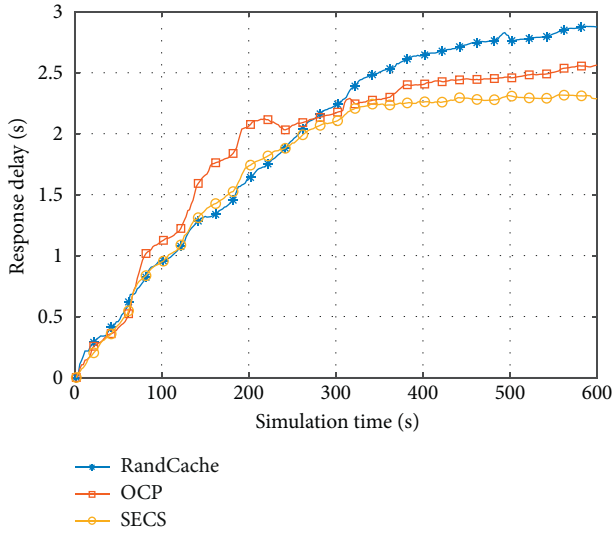

Figure 6: Response delay against simulation time when caching space is 20.

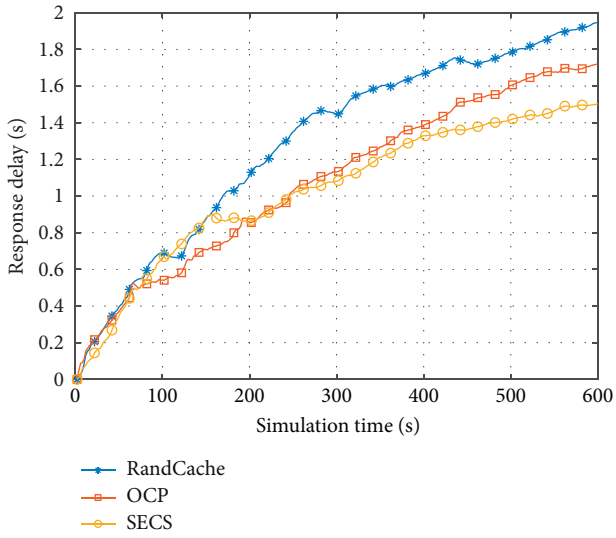

Figure 7: Response delay against simulation time when caching space is 40.

and also a higher QoE due to the low start up delay on video playback. Figures 6 and 7 show the response delay of three solutions when the caching space is 20 and 40, respectively. In both figures, the curves corresponding to all solutions first reveals a fast growing trend before 300 s and then maintains relatively stable. This is because, with more and more users beginning to request video, one mobile node may need to simultaneously serve several users which increases the response delay. After 300 s, due to the number of users that joining equals the number of users quitting the system, the system load and response delay remain stable. Comparing the two figures, we also find that higher caching space can provide better response delay, since larger caching space achieves a higher cache hit ratio, which thereby increases the probability of accessing content from a nearby mobile node. When in stable phase, SECS achieves the lowest RD among three

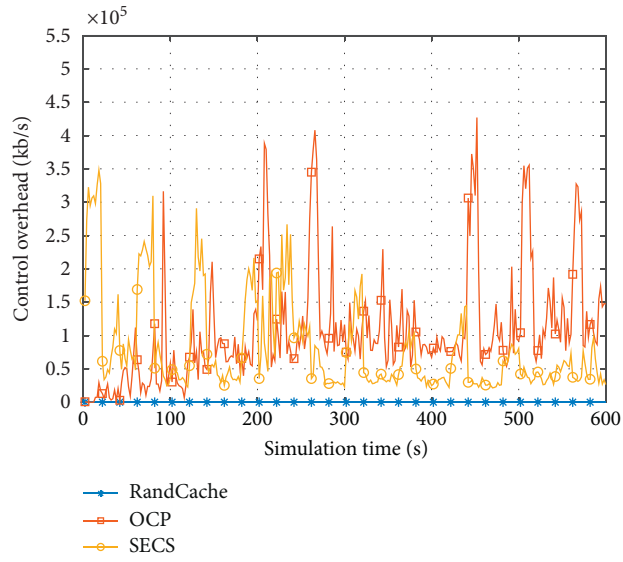

Figure 8: Control overhead against simulation time.

solutions both in the condition of caching spaces 20 and 40. Especially, when caching space is 40, SECS is 11% and 24% lower than OCP and random cache, respectively.

According to Figures 2 and 3, SECS has the highest caching hit ratio, namely, most of video requests can be responded by the mobile nodes nearby the users and hence reduces the response delay. Similarly, OCP achieves the lower RD than random cache because of the similar reason. Random cache policy randomly decide the caching content according to a pregiven fixed probability, whose caching hit ratio can be not guaranteed. Frequent caching miss not only forces the users to access content from distance but also results frequent caching replacement, which further reduce the cache hit ratio. Therefore, random cache performs the worst in terms of the response delay.

Control overhead (CO): in the simulation, we calculate the CO by the averaged bandwidth occupied by delivering the control message of making caching decision. In our SECS, the CO is mainly generated by the information about sharing capacity and interest domain. In OCP, the CO mainly relies on the exchange frequency of mobile node state list and caching decision broadcasting message. In random cache, because each mobile node performs the caching operation based on a fixed probability, there is no control message to exchange. In Figure 8, the curve corresponding to SECS is relatively lower than that of OCP. The main reason is the OCP requires exchange state list which is large in size and more frequent message exchange for accurate predicting the demand variation of the whole system. Although random cache has no CO, this comes at the cost in terms of sacrificing the caching performance including caching hit ratio, caching cost, and response delay.

6. Conclusion

In this paper, we propose a novel Social-aware Edge Caching Strategy of Video Resources in 5G Ultra-Dense Network (SECS). SECS employs the Spectral Clustering to generate initial video clusters and makes use of the Fuzzy C-Means

(FCM) to refine the initial video clusters. The refined video clusters are denoted as the interest domain of users. SECS further makes use of estimating similarity levels of interest domain to cluster the users with common and similar interests into the same groups. By estimating influence for the intragroup sharing performance, SECS designs a performance-aware video caching strategy, which enables the users intelligently implements caching and removing of local video resources to continually optimize video distribution and effectively supports video traffic edge offloading. Extensive tests show how SECS achieves better results in comparison with other state-of-the-art solution OCP in terms of caching hit ratio, caching cost, response delay, and control overhead.

Data Availability

The data used to support the findings of the study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Science and Technology Key Project of Henan Province under Grant nos. 182102210105 and 202102210172, the Training Plan for Young Backbone Teachers of Colleges and Universities in Henan under Grant nos. 2020GGJS191 and 2017GGJS135, the Program for Science and Technology Innovation Outstanding Talents in the Henan Province under Grant no. 184200510011, the Science and Technology Opening Up Cooperation Project of Henan Province under Grant no. 152106000048, the National Natural Science Foundation of China (NSFC) under Grant no. 61501216, and Open Foundation of the Guangxi Key Laboratory of Trusted Software (Grant no. KX202040).

References

[1] D. Ghadiyaram, J. Pan, and A. C. Bovik, "A subjective and objective study of stalling events in mobile streaming videos," *Institute of Electrical and Electronics Engineers Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 183–197, 2019.

[2] M. Zhang, M. Yang, Q. Wu et al., "Smart perception and autonomic optimization: a novel bio-inspired hybrid routing protocol for MANETs," *Future Generation Computer Systems*, vol. 81, pp. 505–513, 2019.

[3] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks," *Institute of Electrical and Electronics Engineers Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2193–2205, 2013.

[4] S. Jia, C. Xu, J. Guan, H. Zhang, and G.-M. Muntean, "A novel cooperative content fetching-based strategy to increase the quality of video delivery to mobile users in wireless networks," *Institute of Electrical and Electronics Engineers Transactions on Broadcasting*, vol. 60, no. 2, pp. 370–384, 2014.

[5] N. Eswara, S. Ashique, A. Panchbhai et al., "Streaming video QoE modeling and prediction: a long short-term memory approach," *Institute of Electrical and Electronics Engineers Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 661–673, 2020.

[6] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "QoE-driven user-centric VoD services in urban multihomed P2P-based vehicular networks," *Institute of Electrical and Electronics Engineers Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2273–2289, 2013.

[7] C. Xu, Z. Li, L. Zhong, H. Zhang, and G.-M. Muntean, "CMT-N. C.: CMT-NC: improving the concurrent multipath transfer performance using network coding in wireless networks," *Institute of Electrical and Electronics Engineers Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1735–1751, 2016.

[8] C. Xu, S. Jia, L. Zhong, H. Zhang, and G.-M. Muntean, "Ant-Inspired mini-community-based solution for video-on-demand services in wireless mobile networks," *Institute of Electrical and Electronics Engineers Transactions on Broadcasting*, vol. 60, no. 2, pp. 322–335, 2014.

[9] F. Rinaldi, S. Pizzi, A. Orsino, A. Iera, A. Molinaro, and G. Araniti, "A novel approach for MBSFN area formation aided by D2D communications for eMBB service delivery in 5G NR systems," *Institute of Electrical and Electronics Engineers Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2058–2070, 2020.

[10] I. Ioannou, V. Vassiliou, C. Christophorou, and A. Pitsillides, "Distributed artificial intelligence solution for D2D communication in 5G networks," *Institute of Electrical and Electronics Engineers Systems Journal*, vol. 14, no. 3, pp. 4232–4241, 2020.

[11] L. Feng, Z. Yang, Y. Yang, X. Que, and K. Zhang, "Smart mode selection using online reinforcement learning for VR broadband broadcasting in D2D assisted 5G HetNets," *Institute of Electrical and Electronics Engineers Transactions on Broadcasting*, vol. 66, no. 2, pp. 600–611, 2020.

[12] R. Zhang, S. Jia, Y. Ma, and C. Xu, "Social-aware D2D video delivery method based on mobility similarity measurement in 5G ultra-dense network," *Institute of Electrical and Electronics Engineers Access*, vol. 8, pp. 52413–52427, 2020.

[13] H. Zhang, S. Chong, X. Zhang, and N. Lin, "A deep reinforcement learning based D2D relay selection and power level allocation in mmWave vehicular networks," *Institute of Electrical and Electronics Engineers Wireless Communications Letters*, vol. 9, no. 3, pp. 416–419, 2020.

[14] M. Sun, X. Xu, X. Tao, P. Zhang, and V. C. M. Leung, "NOMA-based d2d-enabled traffic offloading for 5G and beyond networks employing licensed and unlicensed access," *Institute of Electrical and Electronics Engineers Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4109–4124, 2020.

[15] A. Mehrabi, M. Siekkinen, and A. Yla-Jaaski, "QoE-traffic optimization through collaborative edge caching in adaptive mobile video streaming," *Institute of Electrical and Electronics Engineers Access*, vol. 6, pp. 52261–52276, 2018.

[16] K. Thar, T. Z. Oo, Y. K. Tun, D. H. Kim, K. T. Kim, and C. S. Hong, "A deep learning model generation framework for virtualized multi-access edge cache management," *Institute of Electrical and Electronics Engineers Access*, vol. 7, pp. 62734–62749, 2019.

[17] W. Jiang, G. Feng, S. Qin, and Y. Liu, "Multi-agent reinforcement learning based cooperative content caching for mobile edge networks," *Institute of Electrical and Electronics Engineers Access*, vol. 7, pp. 61856–61867, 2019.

[18] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *Institute of Electrical and Electronics Engineers Internet of Things Journal*, vol. 7, no. 1, pp. 247–257, 2020.

[19] J. Kwak, Y. Kim, L. B. Le, and S. Chong, "Hybrid content caching in 5G wireless networks: cloud versus edge caching," *Institute of Electrical and Electronics Engineers Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3030–3045, 2018.

[20] Y. Liang, J. Ge, S. Zhang, J. Wu, Z. Tang, and B. Luo, "A utility-based optimization framework for edge service entity caching," *Institute of Electrical and Electronics Engineers Transactions on Parallel and Distributed Systems*, vol. 30, no. 11, pp. 2384–2395, 2019.

[21] T. Zhang, X. Fang, Y. Liu, and A. Nallanathan, "Content-centric mobile edge caching," *Institute of Electrical and Electronics Engineers Access*, vol. 30, no. 11, pp. 11722–11731, 2019.

[22] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, D. Niyato, and D. I. Kim, "Distributed deep learning at the edge: a novel proactive and cooperative caching framework for mobile edge networks," *Institute of Electrical and Electronics Engineers Wireless Communications Letters*, vol. 8, no. 4, pp. 1220–1223, 2019.

[23] T. Zhang, X. Fang, Y. Liu, G. Y. Li, and W. Xu, "D2D-Enabled mobile user edge caching: a multi-winner auction approach," *Institute of Electrical and Electronics Engineers Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 12314–12328, 2019.

[24] S. Zhang, W. Sun, and J. Liu, "Spatially cooperative caching and optimization for heterogeneous network," *Institute of Electrical and Electronics Engineers Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11260–11270, 2019.

[25] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *Institute of Electrical and Electronics Engineers Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1791–1805, 2018.

[26] J. Liu, D. Li, and Y. Xu, "Collaborative online edge caching with bayesian clustering in wireless networks," *Institute of Electrical and Electronics Engineers Internet of Things Journal*, vol. 7, no. 2, pp. 1548–1560, 2020.

[27] Y. Guan, X. Zhang, and Z. Guo, "PrefCache: edge cache admission with user preference learning for video content distribution," *Institute of Electrical and Electronics Engineers Transactions on Circuits and Systems for Video Technology*, vol. 2020, Article ID 3006388, 1 page, 2020.

[28] Q. Xu, Z. Su, and R. Lu, "Game theory and reinforcement learning based secure edge caching in mobile social networks," *Institute of Electrical and Electronics Engineers Transactions on Information Forensics and Security*, vol. 15, pp. 3415–3429, 2020.

[29] D. Huang, C.-D. Wang, J.-S. Wu, J.-H. Lai, and C.-K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *Institute of Electrical and Electronics Engineers Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1212–1226, 2020.

[30] Y. Pang, J. Xie, F. Nie, and X. Li, "Spectral clustering by joint spectral embedding and spectral rotation," *Institute of Electrical and Electronics Engineers Transactions on Cybernetics*, vol. 50, no. 1, pp. 247–258, 2020.

[31] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 16–22, 2011.

[32] Y. Li, X. Zhang, X. Li, Y. Zhang, J. Yang, and Q. He, "Mobile phone clustering from speech recordings using deep representation and spectral clustering," *Institute of Electrical and Electronics Engineers Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 965–977, 2018.

[33] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: a framework for in-sample and out-of-sample spectral clustering," *Institute of Electrical and Electronics Engineers Transactions on Neural Networks*, vol. 22, no. 11, pp. 1796–1808, 2011.

[34] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: single-view to multi-view," *Institute of Electrical and Electronics Engineers Transactions on Image Processing*, vol. 25, no. 6, pp. 2833–2843, 2016.

[35] O. Linda and M. Manic, "General type-2 fuzzy C-means algorithm for uncertain fuzzy clustering," *Institute of Electrical and Electronics Engineers Transactions on Fuzzy Systems*, vol. 20, no. 5, pp. 883–897, 2012.

[36] L. Zhong and S. Jia, "Cloud-assisted scalable video delivery solution over mobile ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–10, Article ID 205106, 2015.

[37] C. Xu, M. Wang, X. Chen, L. Zhong, and L. A. Grieco, "Optimal information centric caching in 5G device-to-device communications," *Institute of Electrical and Electronics Engineers Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2114–2126, 2018.

[38] W. Li, "Random texts exhibit Zipf's-law-like word frequency distribution," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 38, no. 6, pp. 1842–1845, 1992.

[39] Q. Li, Y. Zhang, A. Pandharipande, X. Ge, and J. Zhang, "D2D-Assisted caching on truncated Zipf distribution," *Institute of Electrical and Electronics Engineers Access*, vol. 7, pp. 13411–13421, 2019.

[40] L. Gao, G. Zhou, J. Luo, and Y. Huang, "Word embedding with zipf's context," *Institute of Electrical and Electronics Engineers Access*, vol. 7, pp. 168934–168943, 2019.