2022 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-4266-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCNC51071.2022.9771940

Hierarchical Cooperative Caching Strategy in Cached-Enabled Heterogeneous Networks

Dapeng Wu*, Lin Yang*, Yaping Cui*, Peng He*, Ruyan Wang*
*School of Communication and Information Engineering,
Chongqing University of Posts and Telecommunications, Chongqing, China
*Advanced Network and Intelligent Connection Technology Key Laboratory of
Chongqing Education Commission of China, Chongqing, China.
*Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing, China.
Email: wudp@cqupt.edu.cn, s190131045@stu.cqupt.edu.cn, cuiyp@cqupt.edu.cn,
hepeng@cqupt.edu.cn, wangry@cqupt.edu.cn

***Abstract*—The ever-increasing user requests for video services have posed a great challenge to the cellular networks, and the emergence of various new services also puts forward higher requirements to the mobile networks. By caching video contents in cache-enabled heterogeneous networks, the delivery delay of content and the stress of backhaul links can be improved conspicuously. However, how to store diverse contents has got much attentions in the past decade. In this paper, considering the time-varying user requests, a hierarchical cooperative caching strategy with user preference is proposed. Firstly, the caching of content is modeled as a delay optimization problem. Secondly, the historical request data is used to predict the user preference, and the singular value decomposition (SVD) model is further used to predict the missing rating data. Thirdly, both the user preference and rating matrix are used to optimize the caching strategy. Finally, the proposed caching strategy is validated using the MovieLens dataset, the results reveal that the proposed strategy improves the delay performance by at least 35.3% compared with the benchmark strategies.**

***Index Terms*—Cooperative caching, heterogeneous network, user preference, time-varying popularity.**

I. Introduction

The number of equipment connected to the mobile networks will far exceed the global population by 2023 [1]. This trend poses a great challenge to the existing network technology. To accommodate the ever-increasing data requests, densely deployed heterogeneous networks (HetNets) have been regarded as an attractive solution. In this case, the content delivery delay and redundant traffic in the network can be decreased, further, the energy consumption of the network can be saved. Considering the constraints of costs and locations, small base stations (SBSs) usually use their limited backhaul capacity to communicate with core network which can be the bottleneck of the wireless networks. Moreover, the increased density of SBSs incurs the backhaul problem more seriously. Thus, it is urgent to design an efficient caching strategy to mitigate the backhaul burden.

This work was partially supported by Natural Science Foundation of China (Grant 61901070, 61801065, 61771082, 61871062, U20A20157), in part by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant KJQN202000603 and Grant KJQN201900611), in part by the Natural Science Foundation of Chongqing (Grant cstc2020jcyj-zdxmX0024) and in part by University Innovation Research Group of Chongqing (Grant CXQT20017), in part by the General Project of Natural Science Foundation of Chongqing (Grant cstc2021jcyj-msxmX0892).

In the cache-enabled HetNets, caching popular contents in the edge node, the content can be distributed directly through the local SBS instead of the remote content server. Therefore, the response time of user-requested content can be reduced significantly. However, how to cache popular content has an important impact on caching performance. The caching strategy will affected by the content popularity, cache size, the number of adjacent SBSs, and the capacity of MBS(Macro Base Station). Therefore, it is very important to design an effective caching strategy.

There exist many studies on different performance metrics of caching system. A hybrid caching strategy has been employed to maximize the successful transmission probability under the limited backhaul capacity in [2]. In [3], a caching strategy based on the predicted content popularity has been used to maximize the probability of successful content delivery. In [4], a method by jointly considering the cache size allocation and beamforming has been designed to maximize the expected file download rate. A scheme based on content placement, power control and user association has been proposed to tradeoff the energy consumption and file delivery delay in [5].

By considering the user preferences on caching strategy, recent numerous works have turned to exploring content popularity to design sophisticated caching strategies. In [6], by taking user mobility into content preference, a distributed iterative scheme has been used to minimize the sum of communication costs. A continuous proactive caching strategy has been proposed to tradeoff between the energy efficiency and caching hit ratio in [7]. In [8], a distributed caching strategy based on decentralized learning automaton and content popularity has been designed to reduce the download delay.A caching strategy using the matching game has been utilized to minimize the downloading latency and maximize the social welfare simultaneously in [9].

In this paper, We study the content delivery delay minimization problem in cache-enabled HetNets by jointly considering the impact of the time-varying content preferences and the

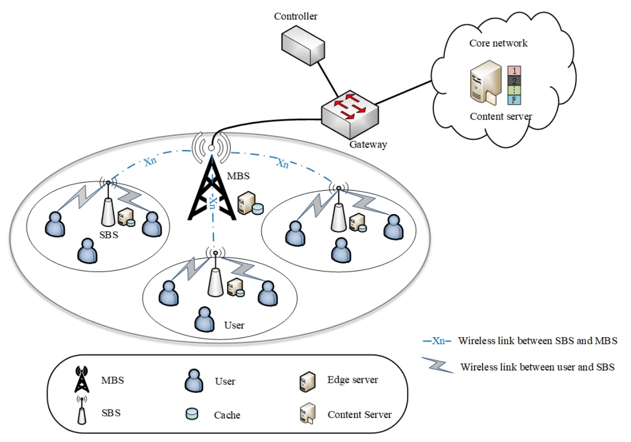

Fig. 1. The cached-enabled HetNets architecture.

limited cache capacity. To efficiently solve the caching problem, we use the real dataset to predict the content preference of users firstly, then we using the predicted user preferences to propose a hierarchical cooperative caching strategy consisting of three phases. Simulation results manifest that the proposed caching strategy can not only exactly predict user preferences but also track user local content requests in real-time. Simulation results also indicate that the proposed strategy possesses the superior performance compared with benchmark strategies and can dramatically decrease the content delivery delay.

The rest of this paper is organized as follows. In Section II, we represent the system model. The delay optimization problem is formulated in Section III. In Section IV, we propose a hierarchical cooperative caching strategy to optimize the content delivery delay. In Section V, we evaluate the performance of our proposed caching strategy through extensive simulations. We conclude this paper in Section VI.

II. System Model

In this section, the network model used in this paper is elaborated firstly, then the caching model is described in detail.

A. Network Model

A three-tier cache-enabled HetNets consisting of MBS, SBSs and users are considered, as illustrated in Fig. 1. In the coverage of the HetNets, the MBS connects to the content server through the gateway to provide service for the N SBSs, SBSs are connected to the MBS via a wireless link to provide service for M users, the user set served by SBS n is M_n^s. We assume that the controller can recognize and record the user requests and the cache state of the base stations (BSs), further the controller can learn the parameters to update the contents. The requested contents are indexed by a set $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$. For content k, its size is denoted by s_k. Let $\mathcal{J}$ denote the BS index and $\mathcal{J} = \{J_1, \ldots, J_n, \ldots, J_N, \ldots, J_{N+1}\}$, where J_{N+1} represents the MBS, J_1 is the first SBS, J_N is the Nth SBS and so on. Each base station is equipped with an edge server with a certain storage capacity. The cache size of SBS and MBS are defined as c_1 and c_2 respectively. The continuous time is split into independent slots $t \in 1, \ldots, T$, where T represents the finite time slots. To characterize the request behavior of the user, each request can be associated with one or more genres. The genres are indexed by a set $\mathcal{Z} = \{z_1, \ldots, z_z\}$. If the content is a member of type 1, $z_1 = 1$, otherwise, $z_1 = 0$. We assume that all users have rated the requested contents according to their preferences.

B. Caching Model

In cache-enabled HetNets, the caching process can be typically divided into the content placement and the content delivery phase. The content placement mainly determines the contents cached in the BSs. In the content placement phase, the controller learns the user preference to optimize the caching strategy. The content delivery mainly determines where to obtain the requested content. In the content delivery phase, the BS (i.e., MBS or SBSs) cannot cache all contents due to the limited cache capacity. If the requested content is cached at the local SBS, the content can be obtained from the local SBS directly. If the content is not stored in the local SBS but cached at the MBS, the MBS will firstly dispense the requested content to the local SBS through the link between the local SBS and MBS, then the local SBS delivers the content to the user. If both the local SBS and the MBS are not cache the requested content, but the adjacent SBS caches the requested content, the adjacent SBS distributes the content to the MBS through the link of MBS and SBS firstly, then the MBS distributes the content to the local SBS, the local SBS delivers the requested content to the user by the link of local SBS and user finally. If all BSs are not cache the requested content, the MBS fetches the content from the remote content server firstly, then the MBS distributes the content to local SBS, the content is sent to the user by the local SBS finally.

III. Problem Formulation

In this section, the average content delivery delay for the cache-enabled HetNets is formulated, the user preference is considered.

In our network model, the communication between the user and the local SBS is carried out through a wireless link. The delivery delay between SBS n and user m can be written as

$$\tau_{m,J_n} = \frac{s_k}{B\log_2\left(1 + \frac{P_n g_{m,n}}{N_0}\right)}, \tag{1}$$

where s_k is the size of content k, B is the channel bandwidth, P_n is the maximum transmission power of SBS n, $g_{m,n}$ is the channel gain of SBS n to user m , and N_0 is the power of the additive white gaussian noise.

The channel gain between the local SBS n and user m is given by

$$g_{m,n} = \varpi \vartheta_{m,n} \Psi_{m,n} d_{m,n}^{-\rho_1}, \tag{2}$$

where ϖ is the system parameter, $\vartheta_{m,n}$ is the fast fading gain of SBS n to user m , $\Psi_{m,n}$ is the slowing fading gain of SBS n to user m , and ρ_1 is the path loss exponent of SBS to user, $d_{m,n}$ is the distance of SBS n to user m.

Similar to equation (1), the delivery delay of MBS to SBS n is $\tau_{J_n,J_{N+1}} = \frac{s_k}{B\log_2\left(1+\frac{P_{N+1}g_{n,N+1}}{N_0}\right)}$, where P_{N+1} represents the maximum transmit power of MBS, $g_{n,N+1}$ is the channel gain of MBS to SBS n.

Similar to equation (2), the channel gain of MBS to local SBS n is $g_{n,N+1} = \varpi\vartheta_{n,N+1}\Psi_{n,N+1}d_{n,N+1}^{-\rho_2}$, where $\vartheta_{n,N+1}$ denotes the fast fading gain of MBS to SBS n, $\Psi_{n,N+1}$ represents the slowing fading gain of MBS to SBS n, and ρ_2 is the path loss exponent of MBS to SBS n, $d_{n,N+1}$ is the distance of MBS to SBS n.

We consider that MBS is connected to the remote server via the optical fiber. The delivery delay between the MBS and content server is τ_1 [10].

We focus on minimizing the content delivery delay by jointly considering the user preferences in MBS and SBSs under the limited cache capacity, and the optimization problem can be expressed as

$$\begin{aligned} &\min_{x} \frac{1}{\Omega}\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} I_m(k,t)\tau_m(k,t) \\ &\text{s.t. } C1: \sum_{k=1}^{K} s_k x_{n,k}(t) \le c_1 \quad n=\{1,\cdots,N\}, \\ &\qquad C2: \sum_{k=1}^{K} s_k x_{N+1,k}(t) \le c_2, \\ &\qquad C3: x_{n,k}(t)\in\{0,1\} \qquad n\in\{1,\cdots,N,N+1\}, \end{aligned} \tag{3}$$

where $\Omega = \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} I_m(k,t)$ is the total amount of requests in T time slots. $I_m(k,t)$ denotes whether the content k requested by the user m or not at t, and $I_m(k,t)=1$ indicates that content k is requested by the user m at t, otherwise, $I_m(k,t)=0$. $x_{n,k}(t)$ indicates whether the content k is stored in SBS n or not at t, $x_{n,k}(t)=1$ stands for the content k is cached in SBS n at t, otherwise, $x_{n,k}(t)=0$. The total size of contents stored at SBS and MBS cannot outnumber the size constraint of SBS c_1 and MBS c_2 in $(C1)$ and $(C2)$ respectively. $C3$ indicates that whether the BS stores the content or does not store the content. $\tau_m(k,t)$ is the delivery delay of the content k demanded by the user m at t, the specific calculation equation of delivery delay $\tau_m(k,t)$ is shown as follows

$$\begin{cases} \tau_{m,J_n}, & \text{(4a)} \\ \tau_{m,J_n} + \tau_{J_n,J_{N+1}}, & \text{(4b)} \\ \tau_{m,J_n} + \tau_{J_n,J_{N+1}} + \tau_{J_{n'},J_{N+1}}, & \text{(4c)} \\ \tau_{m,J_n} + \tau_{J_n,J_{N+1}} + \tau_1, & \text{(4d)} \end{cases}$$

where (4a) represents the delivery delay of the requested content is obtained directly from the local SBS n, (4b) denotes the delivery delay of the requested content is fetched from the MBS, (4c) refers to the delivery delay of the requested content is obtained from the adjacent SBS n', (4d) expresses the delivery delay of the requested content is acquired from the remote content server.

IV. Hierarchical Cooperative Caching Strategy

In this section, firstly, the missing ratings are predicted by the SVD model which considers the impact of information propagation. Then, the historical data is used to predict the user and content popularity. Finally, the workflow of the update and placement is described.

A. Predict Missing Rating

We assume that the user historical ratings of contents are obtained from controller. The user-content rating matrix is denoted as R, where $r_{m,k}$ represents the rating of user m to content k. To forecast the user ratings for all contents, we turn to the collaborative filtering technology with implicit feedback to predict the missing values. The predicted rating can be expressed as follows [11]

$$\begin{aligned} r_{m,k}\approx\hat{r}_{m,k} = \mu + b_m + b_k + V_k^{*}U_m \\ + \alpha(1 - \prod_{i\in\Gamma_k}(1 - \frac{1}{1+e^{-V_k^{*}y_i}})), \end{aligned} \tag{5}$$

where μ is the average rating of all users, b_m denotes the intrinsic bias of user m, b_k is the intrinsic bias of content k, α is a positive constant, V_k is a D-dimensional vector that expresses the latent factor of content k, U_m is the latent vector of user m, $(\cdot)^{*}$ is the transpose of matrix, Γ_k is the user set rated content k, y_i is the user implicit influence vector in Γ_k.

The SVD model aims to minimize the error between the true and predicted values, which is shown as follows

$$\begin{aligned} L =& \frac{1}{2}\sum_{m}\sum_{k\in F_m}(r_{m,k} - \hat{r}_{m,k})^2 + \frac{\lambda}{2}(\sum_{m}||U_m||^2 \\ &+ \sum_{k}||V_k||^2 + \sum_{i}||y_i||^2 + b_m^2 + b_k^2), \end{aligned} \tag{6}$$

where $||\cdot||$ is the Euclidean norm, λ is a tradeoff parameter, F_m is the content set rated by user m.

In order to solve the equation (6), we resort to the stochastic gradient descent to optimize the parameters [11]. The update process of the parameters is given by

$$\begin{aligned} &U_m \longleftarrow U_m + \eta(\sum_{k\in F_m} e_{m,k}{*}V_k - \lambda{*}U_m), \\ &V_k \longleftarrow V_k + \eta(\sum_{m\in\Gamma_k} e_{m,k}{*}(U_m + \alpha A_k) - \lambda{*}V_k), \\ &b_m \longleftarrow b_m + \eta(\sum_{k\in F_m} e_{m,k} - \lambda{*}b_m), \\ &b_k \longleftarrow b_k + \eta(\sum_{m\in\Gamma_k} e_{m,k} - \lambda{*}b_k), \\ &y_i \longleftarrow y_i + \eta(\alpha{*}\sum_{k\in F_i, m\in\Gamma_k} e_{m,k}{*}\frac{V_k}{1+e^{-V_k^{*}y_i}}{*}Q_k - \lambda{*}y_i), \end{aligned} \tag{7}$$

where A_k and Q_k are two auxiliary variables, $e_{m,k}$ is the rating error of user m to content k, η is the learning rate.

The specific calculation of A_k, Q_k and $e_{m,k}$ is represents as follows

$$\begin{cases} A_k = \sum\limits_{i\in\Gamma_k}([\frac{y_i}{1+e^{-V_k^* y_i}}]*Q_k), \\ Q_k = \prod\limits_{i\in\Gamma_k}(1-\frac{1}{1+e^{-V_k^* y_i}}), \\ e_{m,k} = r_{m,k} - \hat{r}_{m,k}. \end{cases} \tag{8}$$

B. Predict User and Content Popularity

In this paper, we define four kinds of parameters: user activity, content popularity, user genre preference and user genre rating. We consider a $M\times K$ user-content mapping matrix $H=(h_{m,k})^{M\times K}$ to predict the user and content popularity. The binary variable $h_{m,k}$ represents whether user m requested the content k or not, if user m requested the content k, $h_{m,k}=1$, otherwise, $h_{m,k}=0$. Denote $h_m=\sum_{k=1}^{K} h_{m,k}$ as the amount of requests sent by user m, and $h_k=\sum_{m=1}^{M} h_{m,k}$ is the total number of requests sent by all users for content k.

The user activity of user m is defined as $a_m^{'}=\frac{h_m}{M}$. For the sake of the description, we normalized the user activity of user m as $a_m=\frac{a_m^{'}}{\sum_{i=1}^{M} a_i^{'}}$.

The popularity of content k is denoted as $p_k^{'}=\frac{h_k}{K}$, the normalized popularity of content k is $p_k=\frac{p_k^{'}}{\sum_{i=1}^{K} p_i^{'}}$.

The genre preference of user m is defined as $q_{m,z}^{'}=\frac{h_m^z}{h_m}$, where h_m^z is the number of genre z in the historical request of user m, the normalized preference of user m on genre z is $q_{m,z}=\frac{q_{m,z}^{'}}{\sum_{i=1}^{Z} q_{m,i}^{'}}$.

The genre rating of user m is defined as $w_{m,z}^{'}=\sum_{i\in\varepsilon_z} r_{m,i}$, where ε_z is the content set requested by user m that belongs to the genre z, the normalized genre rating of user m to genre z is $w_{m,z}=\frac{w_{m,z}^{'}}{\sum_{i=1}^{Z} w_{m,i}^{'}}$.

C. Update and Place Workflow

We resort to the genre preference to obtain the similarity between users, and the similarity of user i and j can be expressed as follows

$$sim(i,j)=\frac{\sum\limits_{z=1}^{Z}(q_{i,z}-\bar{q_i})(q_{j,z}-\bar{q_j})}{\sqrt{\sum\limits_{z=1}^{Z}(q_{i,z}-\bar{q_i})^2}\sqrt{\sum\limits_{z=1}^{Z}(q_{j,z}-\bar{q_j})^2}}, \tag{9}$$

where $\bar{q_i}$ and $\bar{q_j}$ indicate the mean of genre rating of user i and user j respectively.

The idea to find the most similar user set for a target with the optimal stop theory comes from the fact that we usually ask some friends for advice instead of all friends [12]. In particular, we recommend the contents rated by the most similar user set to the target user m. We denote the recommended contents as $\zeta_m=\{\zeta_{m,1},\zeta_{m,2},...\}$, which are not watched by user m. Then the total content set under SBS n can be got by using the content set ζ, the total content set to be demanded by the users of SBS n is denoted as $\phi_n=\{\phi_{n,1},\phi_{n,2},...\}$. Finally, the content set and cache state of all SBSs are used to obtain the content set of MBS, which is denoted as $\theta_{N+1}=\{\theta_{N+1,1},\theta_{N+1,2},...\}$.

We have obtained the recommended content set ζ. Then we take each content in the collection of the recommended contents to obtain the predicted genre rating. We multiply the rating by five ensuring the same range for two kinds of predicted ratings. The predicted genre rating of user m to the content $\zeta_{m,1}$ is given by

$$\beta_{m,\zeta_{m,1}}=5\sum_{z\in\pi_{\zeta_{m,1}}} w_{m,z}, \tag{10}$$

where $\pi_{\zeta_{m,1}}$ is the genre set that belongs to the content $\zeta_{m,1}$.

We have two predicted ratings at present, then we add these two ratings up and obtain a cumulative score for each content. The total score of user m on content $\zeta_{m,1}$ is expressed as

$$\gamma_{m,\zeta_{m,1}}=r_{m,\zeta_{m,1}}+\beta_{m,\zeta_{m,1}}. \tag{11}$$

The cache probability of content $\phi_{n,1}$ in SBS n by using the user activity, content popularity and the total score of content is given by

$$\xi_{n,\phi_{n,1}}=\sum_{m\in M_n^s} a_m p_{\phi_{n,1}}\gamma_{m,\phi_{n,1}}^{'}, \tag{12}$$

where $\gamma_{m,\phi_{n,1}}^{'}$ is the total score variable, if user m has a total score for the content $\phi_{n,1}$, $\gamma_{m,\phi_{n,1}}^{'}=\gamma_{m,\phi_{n,1}}$, otherwise, $\gamma_{m,\phi_{n,1}}^{'}=0$.

Similar to the equation (12), the cache probability of content $\theta_{N+1,1}$ in MBS can be expressed as

$$\xi_{N+1,\theta_{N+1,1}}=\sum_{m=1}^{M} a_m p_{\theta_{N+1,1}}\gamma_{m,\theta_{N+1,1}}^{'}. \tag{13}$$

Figure 2 is an example of content update. At a given moment, the current contents in the BS are A, B, C and D, which are both 3Mb. The contents requested during this update time are E, F, G, H and I of sizes 3Mb, 2Mb, 3Mb, 2Mb and 1Mb, respectively. The ξ-values of the contents at the start of the update process are shown in the figure. This process mainly replaces the contents by comparing the ξ-values of the current cached contents with the requested contents. The content with a larger ξ-value will be cached firstly. Only the content whose size is smaller than the remaining cache space will be cached due to the limited cache space. We can see from the figure that the content C and D with smaller ξ-values are evicted and the content E and F with larger ξ-values are cached. Owing to the limitation of the remaining space of the cache, the content I is cached finally.

In the content placement phase. Firstly, the content set ϕ_n is cached according to the cache probability ξ_n. Then, the cached contents of MBS θ_{N+1} are found in the light of the content set ϕ and the cache state of SBSs x. Next, the content set θ_{N+1} is cached according to the cache probability ξ_{N+1}. Consequently, the optimal cache state of BS is obtained.

In the content update phase, during each cache update time T_{up}, the requests within the update interval are used to update the user preference. Firstly, the process of content update is

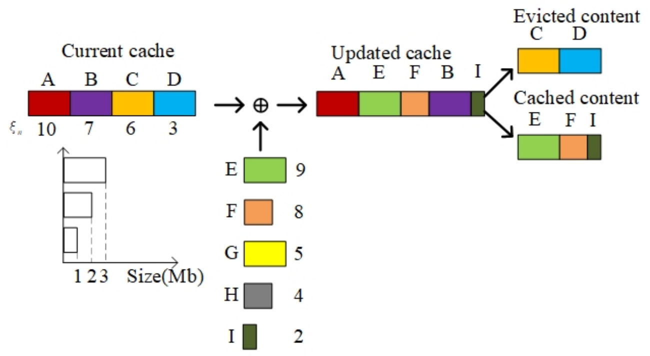

Fig. 2. An example of content update.

used to update the contents of N SBSs. Then, the same content update method is used to find the content set to be cached in the MBS. Finally, the content update process is used to update the contents of MBS.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Channel bandwidth	10 GHz
Transmit power of MBS	43 dBm
Transmit power of SBS	23 dBm
Noise power	-114 dBm
Coverage radius of MBS	1000 m
Coverage radius of SBS	150 m
Dimension of the latent factor vector	20
User implicit impact factor	0.8
Delay between MBS and Content Server	20 s

V. SIMULATION RESULTS

We validate the proposed caching strategy using Python and scikit-learn library, and we choose the movie data from the MovieLens Dataset as the cached contents. Each of the data includes the userID, movieID, rating and timestamp. Besides, the dataset also provides the movie genres. Each movie is related to one or more of the 19 genres, which contain the action, comedy, drama, musical, etc.. The entire process of caching is simulated by splitting the selected data. In our simulation, we consider the HetNets composed of one MBS and three SBSs, and each SBS serves three users. The locations of users and SBSs are randomly generated and deployed. The key parameters of the settings for our simulations are presented in Table I.

We compare the performance with the following caching strategies.

1) Random Caching (RC): In this caching strategy, the contents are cached randomly.

2) Least Recently Used (LRU): In this caching strategy, the most recently requested contents are cached in the memory.

3) Least Frequently Used (LFU): In this caching strategy, the most frequently requested contents are cached in the memory.

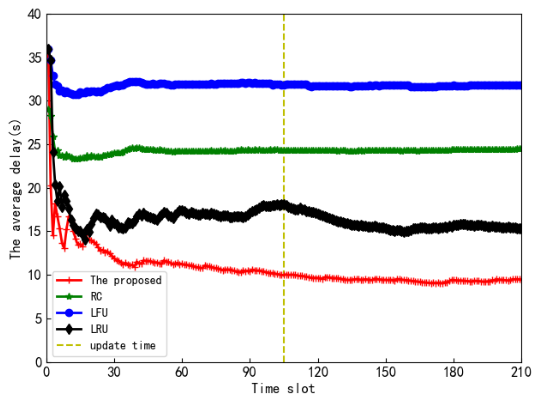

Fig. 3. Time slot vs. the average delay.

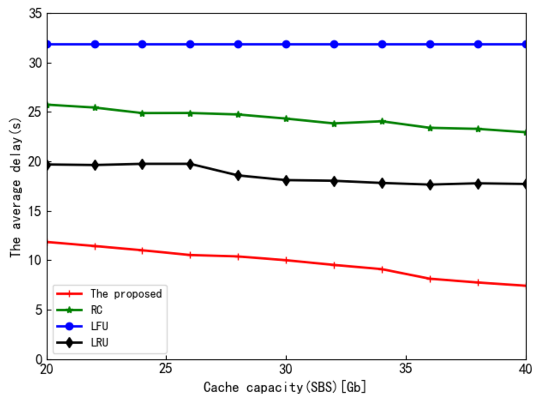

Fig. 4. Cache capacity(SBS) vs. the average delay.

In Fig. 3, we present the variation of the average delay over a finite time T. The figure also includes three baseline strategies (viz. RC, LFU and LRU). The cache capacity of SBS and MBS is 30Gb and 60Gb respectively. The first half of the figure (i.e., the left part of the yellow dotted line) represents the simulation result of the placement phase, and the second half of the figure (i.e., the right part of the yellow dotted line) represents the results of the update phase. We can observe that the average delay of different strategies tends to be stable over time, and the reason is that more user requests will facilitate the accuracy of caching decisions. It can also be noticed that the proposed caching strategy can significantly reduce the average delay of content in both content placement and update phases. Compared with the LRU, RC and LFU, the proposed caching strategy improves the delay performance by 35.3%, 54.2% and 65.5% respectively. The reason is that the proposed caching strategy can predict the popularity of content in real-time.

Figure 4 is the average delay of different cache sizes of SBS. We can find that the average delay of four caching strategies gradually decrease along with the increase of the cache size of SBS, which is due to the more popular contents can be cached in SBS, and users do not need to obtain contents from the remote content server. Moreover, the proposed caching strategy is better than three other strategies. Compared with the three strategies, the delay performance is improved by 44.4%,

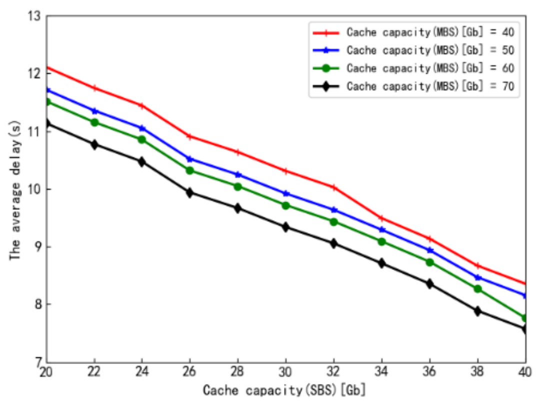

Fig. 5. Cache capacity(MBS) vs. the average delay.

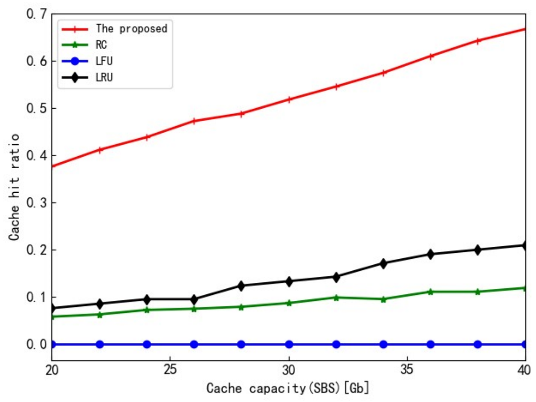

Fig. 6. Cache capacity(SBS) vs. cache hit ratio.

60.0% and 68.7% respectively. But the average delay remains almost stable because of the user requests in the real dataset do not obey the LFU distribution.

The impact of the size of MBS on the average delay is shown in Fig. 5. The number of cached contents in the BS is determined by the cache capacity. The larger the cache space is, the more content can be cached, so the scale of contents can be controlled by adjusting the cache capacity of BS. We can see that the average content delivery delay is decreased with the cache size. The reason is that more popular contents are cached in the MBS, and users do not need to obtain the content from adjacent SBS or remote content server, thereby the average delay can be reduced. For example, when the size of SBS is 30Gb, the average delay drops from 10.5s to 9.4s. We can also observe that, as the cache size of MBS and SBS increases, the average delay decreases. Therefore, the cooperative caching between MBS and SBSs can reduce the average content delivery delay evidently.

In Fig. 6, we show the impact of the cache capacity of SBS on the cache hit ratio. The cache capacity of MBS is 60Gb. It can be seen that the cache hit ratios of RC, LRU and the proposed caching strategy show an upward trend with the growth in the cache size, and the proposed caching strategy is better than other strategies obviously. The proposed caching strategy improves the cache hit ratio by 42.0% and 51.0% compared to RC and LRU respectively. The reason is that the contents are cached by using the local user preference, and the user preference makes the cached contents more suitable. We can also observe that the cache hit ratio of the LFU is almost zero, this is because the user requests in the real dataset do not follow the LFU distribution.

VI. Conclusions

In this paper, we cache the contents interested by users by using the predicted content popularity at the BS in advance, so users can obtain content from nearby service nodes to reduce content distribution delay. Considering the heterogeneous content sizes and time-varying user preferences, we propose a hierarchical cooperative caching strategy. And the proposed caching strategy is validated by using the MovieLens dataset. Simulation results demonstate that compared with three other caching strategies, the proposed caching strategy can improve the average delivery delay and the cache hit ratio significantly. This paper considers the case under the coverage of a single MBS, however, the proposed caching strategy can also be applicable to the multiple MBSs case. The delivery delay can be further reduced through the cooperation between MBSs. In future, we will introduce the coded caching into our work to improve the key performance indicator.

References

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

[2] Y. Cui and D. Jiang, "Analysis and optimization of caching and multicasting in large-scale cache-enabled heterogeneous wireless networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 250-264, 2017.

[3] N. Garg, M. Sellathurai, V. Bhatia, B. N. Bharath and T. Ratnarajah, "Online content popularity prediction and learning in wireless edge caching," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1087-1100, 2020.

[4] B. Dai, Y. Liu and W. Yu, "Optimized base-station cache allocation for cloud radio access network with multicast backhaul," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1737-1750, 2018.

[5] H. Wu, H. Lu, F. Wu and C. W. Chen, "Energy and delay optimization for cache-enabled dense small cell networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7663-7678, 2020.

[6] J. Zhou, X. Zhang and W. Wang, "Social-aware proactive content caching and sharing in multi-access edge networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1308-1319, 2020.

[7] Y. Fang, W. Chen and L. Li, "Balancing energy efficiency and hit ratio in social-aware caching: A cross layer approach," *IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, USA, 2019.

[8] C. Ma et al., "Socially aware caching strategy in device-to-device communication networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4615-4629, 2018.

[9] J. Li et al., "On social-aware content caching for D2D-enabled cellular networks with matching theory," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 297-310, 2019.

[10] X. Li, X. Wang, S. Xiao and V. C. M. Leung, "Delay performance analysis of cooperative cell caching in future mobile networks," *IEEE International Conference on Communications (ICC)*, London, 2015, pp. 5652-5657.

[11] F. Xiong, W. Shen, H. Chen, S. Pan, X. Wang and Z. Yan, "Exploiting implicit influence from information propagation for social recommendation," *IEEE Transactions on Cybernetics*, vol. 50, no. 10, pp. 4186-4199, 2020.

[12] Nyhoff J, "Algorithms to live by: The computer science of human decisions," *Perspectives on Science and Christian Faith*, 2017, 69(2): 127-129.