Graph-Theoretic Approach for Cache Placement and Delay Optimization in Cache-Enabled Mobile Networks

Fang Dong, Tianyu Wang, and Shaowei Wang
School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China
E-mail: mf1623009@smail.nju.edu.cn, {tianyu.alex.wang, wangsw}@nju.edu.cn

***Abstract*—Caching at base stations (BSs) is a promising scheme to alleviate the traffic burden in mobile communication systems. In this paper, we aim to minimize the average delay of all users in the cache-enabled mobile network where the BSs can exchange data with each other via X2 interface. We jointly consider cache placement and user association problems and employ graph theory to deal with the optimization task. For a given network graph, we aim to find the maximum cliques and place different files in the maximum clique so as to improve local cache hit probability. In the user association procedure, we make the BSs which store the requested files of users serve these users as many as possible. Simulation results show that our proposed algorithm yields the lowest delay among the other representative algorithms.**

***Index Terms*—Cache, graph theory, mobile networks.**

I. Introduction

Mobile communications networks transport us towards the era of internet of things (IOT), however, the massive connections generated by the IOT applications also bring a huge challenge to the mobile communications industry. Investigations show that data transmission traffic will increase by 8 times by 2020 compared to 2015 [1, 2]. This trend is due to the increasing demands for a large number of rich media files, such as image and video streaming, which require a large amount of data traffic and cause a serious burden on the network. On the other hand, many requests in the mobile networks are duplicated. If the unnecessary redundant data transmission in the network can be reduced, the heavy network load would be greatly alleviated.

Caching is proposed as a key technology for the LTE networks. In [3], caching is proposed as a service in mobile networks and a competitive mechanism is developed to minimize the average user delay in the competition of different service providers. A novel caching framework is presented for offloading backhaul and fronthaul load systems in the cloud radio access networks in [4]. In [5], the authors first measure the spatial and temporal request patterns in mobile video systems, then a geo-cooperative caching strategy is designed for mobile video transmission. In [6], virtual caching placed at network edges is discussed. As an important issue in cache-enabled networks, cache placement attracts attentions from researchers of both academia and industry. In [7], a greedy algorithm is proposed for content placement in wireless network and the average bit error rate is minimized. In [8], the authors optimize content placement to improve the hit probability in both single-tier and multi-tier heterogeneous cellular network. In [9], a proactive caching policy is proposed to maximize the offload probability with consideration of average energy consumption in D2D communication networks.

It is noticeable that user association issue in cache-enabled networks is also an important problem. In [10], the authors consider both the quality of experience (QoE) requirements of users and routing policies in the video cached network. In [11], a scheme aiming to utilize small base station (BS) is proposed with consideration of bandwidth allocation and backhaul utilization, as well as a cache-aware user association algorithm that minimizes the backhaul consumption of each BS while satisfying the QoE requirements. A delay-based heterogeneous cellular network caching strategy is proposed in [12], where user association is considered together and a distributed algorithm is developed to minimize the average download delay of users.

In this paper, we aim to minimize the average delay of all users in LTE networks, where BSs exchange information via X2 interface. We consider cache placement and user association problems. In the cache placement, we introduce graph theory method. Each vertex in the graph represents each BS. The BSs that can exchange information through X2 interface are connected by edges. We find the maximum cliques of connected vertices in the graph, and place different files at these BSs in the same clique to improve cache hit probability. Due to the limited radio resources in the BSs, we consider the bandwidth and power constraints and associate as many users as possible to the BS where the requested files can be found. The contributions of this paper are as follows:

- We jointly optimize cache placement and user association to minimize the average delay. We propose a graph-theoretic method to store different files in local BSs so that the average download of the files from the core network can be minimized, then an efficient user association algorithm is proposed with consideration of bandwidth and power constraints.
- Our proposal method can significantly improve system

This work was partially supported by the National Natural Science Foundation of China (61671233, 61801208), the Jiangsu Science Foundation (BK20151389, BK20170650), the Postdoctoral Science Foundation of China (BX201700118, 2017M621712), and the Jiangsu Postdoctoral Science Foundation (1701118B).

978-1-5386-6119-2/18/$31.00 ©2018 IEEE

performance as compare to other representative methods. The average delay of users is much less than the most popular caching policy (MC) and the random caching policy (RC) schemes.

The reminder of this paper is organized as follows. In Section II, we present system model and formulate our optimization task. In Section III, we describe the algorithms in detail. Numerical results and discussions are presented in Section IV. Finally, we conclude this paper in Section V.

II. System Model and Problem Formulation

Consider an LTE network architecture with X2 interfaces among the BSs and S1 interfaces between the BSs and the core network. In practical mobile networks, only BSs within a certain range of each other can exchange data via X2. There are $\mathcal{N}$ BSs denoted as set $\mathcal{N} = \{1, 2, 3...N\}$ and $\mathcal{K}$ users denoted as set $\mathcal{K} = \{1, 2, 3...K\}$. The hierarchical cache-enabled mobile network topology is illustrated in Fig. 1, where serving gateway, packet data network gateway and mobility management entity are denoted as S-GW, P-GW and MME, respectively. A finite files library $\mathcal{F} = \{1, 2, 3, ...F\}$ including F files should be requested by users. The size of each file is S_f. BS n has a limited storage capacity CAP_n to cache selected files from $\mathcal{F}$. Each user may request some files f from the library $\mathcal{F}$, $f \in \mathcal{F}$. The probability of user request corresponds the zipf distribution:

$$z_f = f^{-\alpha}(\sum_{f=1}^{F} f^{-\alpha}), \tag{1}$$

where α is zipf parameter, reflecting the skewed distribution of the popularity of content, that is, $\sum_{f=1}^{F} z_f = 1$ [13]. Statistically speaking, the larger the value, the more possible the file to be requested.

The available bandwidth and power of BS n is b_n^{max} and p_n^{max}, respectively. $h_{k,n}$ is the channel gain between user k and BS n. $b_{k,n}$ and $p_{k,n}$ represent the bandwidth and power allocation if BS n serves user k, respectively. For user $k \in \mathcal{K}$, the rate requirement is R_k^{min}. According to Shannon's theorem, the available transmission rate between user k and BS n can be expressed as follows:

$$r_{k,n} = b_{k,n} \log_2 \left[1 + \frac{p_{k,n} h_{k,n}}{b_{k,n}(N_0 + I_{k,n})}\right], \tag{2}$$

where N_0 represents the power spectral density of additive white gaussian noise. $I_{k,n}$ is the interference introduced by other BSs with unit bandwidth:

$$I_{k,n} = \sum_{n^* \in \mathcal{N}, n^* \neq n} \frac{p_{n^*}^{max} h_{k,n^*}}{b_{n^*}^{max}}. \tag{3}$$

Let $\psi_{k,n}$ denote whether the requested file of user k can be found in local BSs or not:

$$\varphi_{k,n} = \begin{cases} 1 & \text{requested file of user } k \text{ is found locally;} \\ 0 & \text{requested file of user } k \text{ is not found locally.} \end{cases} \tag{4}$$

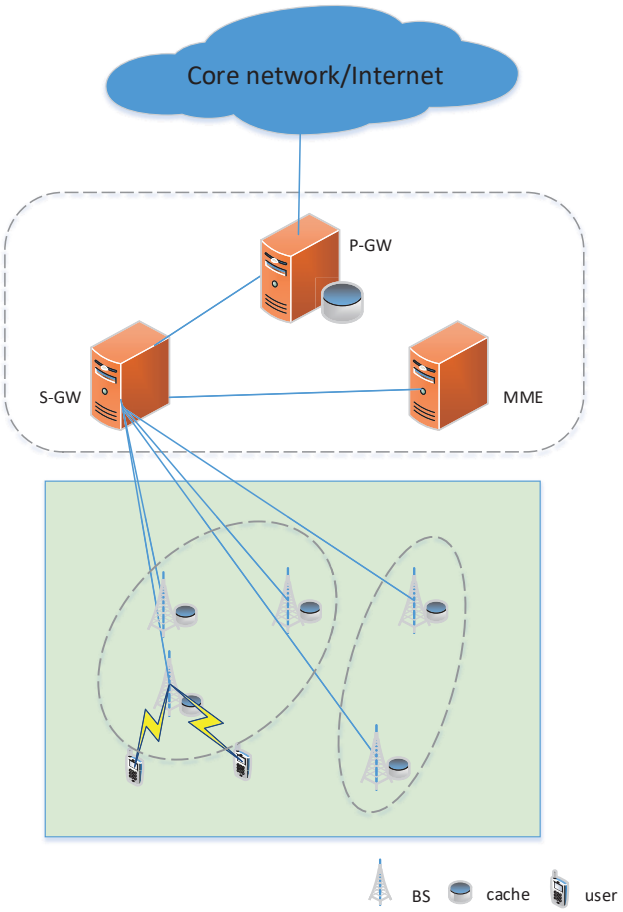

Fig. 1. Illustration of considered system model

Binary variable $\rho_{k,n}$ indicates that whether user k is associated with BS n or not:

$$\rho_{k,n} = \begin{cases} 1 & \text{user } k \text{ associated with BS } n; \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

and $w_{f,n}$ shows whether file f is placed in BS n or not:

$$w_{f,n} = \begin{cases} 1 & \text{file } f \text{ cached in BS } n; \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The delay of user to get files in mobile networks includes wireless transmission delay and backhaul delay. The wireless transmission delay between user k and BS n depends on the size of the requested file and transmission rate, which can be expressed as follows:

$$D_{k,n}^1 = \frac{S_f}{r_{k,n}} = \frac{S_f}{b_{k,n} log_2[1 + \frac{p_{k,n} h_{k,n}}{b_{k,n}(N_0 + I_{k,n})}]}, \tag{7}$$

For wired backhaul, the backhaul delay of BSs is related to the average link distance, the average traffic load. Generally speaking, it can be modeled as a random variable following

exponential distribution with a mean value of D_B [14]. If the requested file can be found locally, there is no backhaul delay can be avoided. We can expressed it as follows:

$$D_{k,n}^2 = (1-\varphi_k)D_B. \tag{8}$$

Consequently, the delay for user k to request file f when served by BS n is described as

$$D_{k,n} = D_{k,n}^1 + D_{k,n}^2 = \frac{S_f}{r_{k,n}} = \frac{S_f}{b_{k,n}log_2[1+\frac{p_{k,n}h_{k,n}}{b_{k.n}(N_0+I_{k,n})}]} + (1-\varphi_k)D_B. \tag{9}$$

Let D be the average delay of all users, thus D can be calculated as

$$D = \frac{1}{|K|}\sum_{k\in\mathcal{K}}\sum_{f\in\mathcal{F}}\sum_{n\in\mathcal{N}}\varphi_{k,n}\rho_{k,n}D_{k,n}. \tag{10}$$

We aim to minimize the average delay of all users while taking into consideration of practical constraints including the bandwidth and power budgets of system. The optimization problem can be formulated as follows:

$$\begin{aligned}
&\min_{\rho_{k,n},b_{k,n},p_{k,n},w_{f,n}} D \\
s.t.\quad & C_1: \sum_{k\in\mathcal{K}} b_{k,n} \le b_n^{max}, \forall n \in \mathcal{N}, \\
& C_2: \sum_{k\in\mathcal{K}} p_{k,n} \le p_n^{max}, \forall n \in \mathcal{N}, \\
& C_3: \sum_{f\in\mathcal{F}} w_{f,n} \le CAP_n, \forall n \in \mathcal{N}, \\
& C_4: r_{k,n} \ge R_k^{min}, \forall k \in \mathcal{K}, \\
& C_5: b_{k,n} \ge 0, p_{k,n} \ge 0, \forall k \in \mathcal{K}, \forall n \in \mathcal{N}, \\
& C_6: \rho_{k,n} \in \{0,1\}, \forall k \in \mathcal{K}, n \in \mathcal{N}, \\
& C_7: w_{f,n} \in \{0,1\}, \forall f \in \mathcal{F}, n \in \mathcal{N}, \\
& C_8: \sum_{n=1}^{N} \rho_{k,n} \le 1, \forall k \in \mathcal{K}, n \in \mathcal{N},
\end{aligned} \tag{11}$$

where the inequalities of C_1 and C_2 are the constraints of bandwidth and power for each BS. C_3 is the cache capacity for each BS. C_4 means the transmission rate requirements. $C_5 \sim C_7$ are intuitive. C_8 denotes each user can only be served by one BS.

III. Our Proposed Algorithms

As mentioned above, our proposed procedure consists of two parts: cache placement procedure and user association. In the cache placement section, we find the sets of BSs in which any two BSs are connected by an edge. Then we put different files in each BS in the set to improve cache hit probability. Then we work out the user association problem to satisfy given number of users with required transmission rate, where a joint bandwidth and power allocation algorithm is introduced.

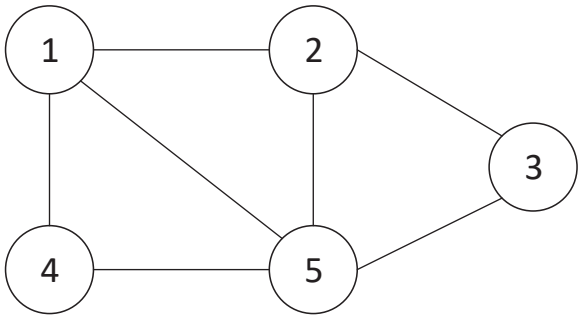

Fig. 2. Illustration of graph model

A. Cache Placement Procedure

Given an undirected graph $G = (V, E)$, where V is the set of vertices which represents BSs and E is the set of edges between two vertices. If two nodes are connected by an edge, it means that they are connected via X2 interface. We find the largest set of vertices where any two vertices in the set are connected by an edge. It is the maximum clique of graph. For example, try to find a maximum clique with five vertices in the graph shown in Fig. 2. There are three maximum cliques in this graph, which is $\{1,2,5\}, \{1,4,5\}, \{2,3,5\}$.

As an integer programming problem, maximum clique problem (MCP) has many equivalent descriptions, the integer programming problem is described below: Let $t: (0,1)^n \longrightarrow 2^v, t(x) = \{i \in V : x_i = 1\}, \forall x \in \{0,1\}^n, \forall S \in 2^v$, then $x = t^{-1}(S) = \{x_i : i = 1,2,...,n\}$, n is the vertex number of a graph, where

$$x_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases} \tag{12}$$

$$\begin{aligned}
&\min f(x) = -\Sigma_{i=1}^n x_i \\
s.t.\quad & x_i + x_j \le 1, \forall (i,j) \in E', x \in \{0,1\}^n,
\end{aligned} \tag{13}$$

if x^* is the optimal solution to (13), then the set of $C = t(x^*)$ is one of the largest clumps of graph G and $|C| = -f(x^*)$. Due to $x_i, x_j \in \{0,1\}, x_i + x_j \le 1, \forall (i,j) \in E'$ if and only if $x_i x_j = 0$, we have

$$f(x) = -\sum_{i=1}^{n} x_i + 2\sum_{(i,j)\in E', i>j} x_i xj = x^T(A_G - I)x, \tag{14}$$

where $A_{G'}$ is the adjacency matrix of complementary graph G' of graph G. Maximum clique problem is equivalent to the global quadratic 0/1 problem:

$$\begin{aligned}
&\min f(x) = x^T Ax \\
s.t.\quad & x \in \{0,1\}^n
\end{aligned} \tag{15}$$

where $A = A_{G'} - I$. If x^* is the optimal solution to (15), the set of $C = t(x^*)$ is one of the maximum group in graph G and $|C| = -f(x^*)$.

TABLE I
MAXIMAL CLIQUE PROCEDURE

Algorithm 1

Given the matrix of the graph, initialization R, P, X
MCP (R, P, X)
  if P and X are both empty
    report R as maximal clique
  for each vertex v in P
    MCP $(R \cup \{v\}, P \cap N(v), X \cap N(v))$
    $P := P \backslash \{v\}$
    $X := X \cup \{v\}$

The algorithm to find maximum cliques in a given network graph is given in Table I. Let R be the set that records vertices that have been added to current maximum clique and set P records vertices that have the possibilities to be added in set R, that is, vertices that may have edges with all points in R. Set X records the vertices that have completed the maximum clique count, which role is to judge the repeatability for each vertex. $N(v)$ is a set of adjacent vertices of vertex v. The procedure is as follows: Initialize set R and X to be empty and P is the set of all vertices. Select a vertex v from set P, add vertex v into set R and move the vertex which is not in $N(v)$ out of the set P and X. Then select another vertex from the remaining P and repeat the above operation until P is an empty set. If set X is also empty at this time, then R is a new maximum clique, if set X is not empty, then R is a subset of a maximum clique that has been found. Then back to the last selected vertex v in set P and restore the set R, P, X to the original states, meanwhile, remove the selected vertex v from set P and add it to set X. Repeat the above operation until all vertices have been traversed. After that, we put different files in each maximum clique group to improve local cache hit probability.

B. Delay Minimization Procedure

1) Bandwidth and power allocation algorithm: According to Eq. (2), if all bandwidth of BS n are allocated to serve user set $\mathcal{K}_n$, the required power for each user can be worked out:

$$p_{k,n} = \frac{b_{k,n}}{H_{k,n}} \left(2^{\frac{R_k^{min}}{b_{k,n}}} - 1 \right). \tag{16}$$

Thus, we should solve the following optimization problem:

$$\begin{aligned} \min_{b_{k,n}} \quad & \sum_{k \in \mathcal{K}_n} \frac{b_{k,n}}{H_{k,n}} \left(2^{\frac{R_k^{min}}{b_{k,n}}} - 1 \right) \\ s.t. \quad C_1: \quad & \sum_{k \in \mathcal{K}_n} b_{k,n} = b_n^{max}, \\ C_2: \quad & b_{k,n} \geq 0, \forall k \in \mathcal{K}_n. \end{aligned} \tag{17}$$

Eq. (17) turns to be a convex problem that can be solved by convex optimization methods. The Lagrangian of (17) is

$$\begin{aligned} L = & \sum_{k \in \mathcal{K}_n} \frac{b_{k,n}}{H_{k,n}} \left(2^{\frac{R_k^{min}}{b_{k,n}}} - 1 \right) \\ & + \lambda \Big(\sum_{k \in \mathcal{K}_n} b_{k,n} - b_n^{max} \Big) - \sum_{k \in \mathcal{K}_n} \mu_{k,n} b_{k,n}, \end{aligned} \tag{18}$$

TABLE II
BANDWIDTH AND POWER ALLOCATION

Algorithm 2

Initialization: $Cnt = 0, \lambda^{(Cnt)} = 0, \lambda_{min} = 0, \lambda_{max} = \Gamma$;
repeat
  $Cnt = Cnt + 1$;
  $\lambda^{(Cnt)} = (\lambda_{min} + \lambda_{max})/2$;
  for $k \in \mathcal{K}_n$
    Calculate $b_{k,n}$ that satisfies Eq. (17);
    $b_{k,n}$=max$\{0, b_{k,n}\}$;
  end for
  if $\sum_{k \in \mathcal{K}_n} b_{k,n} > b_n^{max}$
    $\lambda_{min} = \lambda^{Cnt}$;
  else
    $\lambda_{max} = \lambda^{Cnt}$;
  end if
until $|\lambda^{(Cnt)} - \lambda^{(Cnt-1)}| \leq \epsilon$;
for $k \in \mathcal{K}_n$
    $b_{k,n}^* = b_{k,n}$;
    Calculate $p_{k,n}^*$ using Eq. (16);
end for
return $b_{k,n}^*, p_{k,n}^*, \sum_{k \in \mathcal{K}_n} p_{k,n}^*$.

λ and $\mu_{k,n}$ are Lagrange multipliers. Denote $b_{k,n}^*$ and λ^*, $\mu_{k,n}^*$ be primal and dual optimal points with zero duality gap. According to Karush-Kuhn-Tucker conditions [13], we need to solve the following equations:

$$\lambda^* = -\frac{1}{H_{k,n}} \left[\left(1 - \frac{R_k^{min} \ln 2}{b_{k,n}^*} \right) 2^{\frac{R_k^{min}}{b_{k,n}^*}} - 1 \right], \tag{19}$$

$$\sum_{k \in \mathcal{K}_n} b_{k,n}^* = b_n^{max}, \tag{20}$$

$$\mu_{k,n}^* = 0, b_{k,n}^* > 0. \tag{21}$$

$b_{k,n}^*$ and λ^* can be obtained by using bisection method. The detail of bandwidth and power allocation algorithm is in Table II, where ϵ and Γ are a tolerance and a appropriate positive integer, respectively. Suppose $P_n(\mathcal{K}_n) = \sum_{k \in \mathcal{K}_n} p_{k,n}^*$ as the optimal value of (17). If $P_n(\mathcal{K}_n)$ is less than the power budget of BS n, we claim that BS n can serve $\mathcal{K}_n$ users with given rate requirements.

2) User Association Algorithm: Assume K users with rate requirements should be served by N BSs with limited bandwidth and power, we need to decide each user should be served by which BS. The required power $p_n(\{k\})$ for user k served by BS n is as follows:

$$p_n(\{k\}) = \frac{b_{k,n}}{H_{k,n}} \left(2^{\frac{R_k^{min}}{b_{k,n}}} - 1 \right), \tag{22}$$

where $H_{k,n} = \frac{h_{k,n}}{(N_0 + I_{k,n})}$. Initialize $\mathcal{K}_n = \varnothing$, $\mathcal{K}_{temp} = \mathcal{K}$ and $\mathcal{N}_{temp} = \mathcal{N}$. Define $\mathcal{K}_{n'}$ as the set of users that have been served. The remaining users and candidate BSs are denoted by $\mathcal{K}_{temp}$ and $\mathcal{N}_{temp}$, respectively. We calculate $p_n(\{k\})$ for $k \in \mathcal{K}_{temp}$, $n \in \mathcal{N}_{temp}$ in each loop and work out the index (k', n') in the light of the lowest required power $p_{n'}(\{k'\})$, then we calculate $p_{n'}(\mathcal{K}_{n'} \cup \{k'\})$. If $p_{n'}(\mathcal{K}_{n'} \cup \{k'\}) \leq p_n^{max}$,

TABLE III
USER ASSOCIATION PROCEDURE

Algorithm 3

Initialization: $\mathcal{K}_n = \varnothing, \forall n \in \mathcal{N}$; $\mathcal{K}_{temp} = \mathcal{K}$; $\mathcal{N}_{temp} = \mathcal{N}$;
Calculate $p_n(\{k\}), k \in \mathcal{K}, n \in \mathcal{N}$;
repeat
  $(k', n') = \arg\min_{(k,n):k\in\mathcal{K}_{temp}, n\in\mathcal{N}_{temp}} p_n(\{k\})$;
  if $p_{n'}(\mathcal{K}_{n'} \cup \{k'\}) \leq p_{n'}^{max}$
    $\mathcal{K}_{n'} \leftarrow \mathcal{K}_{n'} \cup \{k'\}$;
    $\mathcal{K}_{temp} \leftarrow \mathcal{K}_{temp} \backslash \{k'\}$;
  else
    $\mathcal{N}_{temp} \leftarrow \mathcal{N}_{temp} \backslash \{n'\}$;
  end if
until $\mathcal{K}_{temp} = \varnothing$ or $\mathcal{N}_{temp} = \varnothing$
return $\mathcal{K}_n$

it means that the user k' can be associated with BS n, thus we add k' into K_{temp}, $K_{temp} = K_{temp} \backslash \{k'\}$. Otherwise, BS n' cannot satisfy the rate requirements of the remaining users because user k' requires the least power as compared to remaining if consuming the same amount of bandwidth. Consequently, BS n' should be removed from $\mathcal{N}_{temp}$. This procedure terminates on condition that all users have been served by the BSs or all active BSs cannot serve any users. Our proposed user approximation algorithm **(Algorithm 3)** is described in Table III.

Theorem 1. *Algorithm 3 is a* $\frac{1}{2}$*-approximation algorithm for user association problem.*

The proof can be found in [15, 16].

IV. NUMERICAL RESULTS

We compare our algorithm with other representative cache algorithms: most popular files cached at BSs (MC) and randomly placing files at BSs (RC). Our proposal is labeled as OC. The system parameters are as follows: The bandwidth of each BS is $10MHz$ and the maximum transmission power of BS is 1 W. The path loss model is based on 3GPP standard and expressed as $140.7 + 36.7\log_{10}(D)$, where D (in km) is the distance between BSs and users. The standard deviation of lognormal shadowing and the value of noise power spectral density (PSD) are 10 dB and -184 dBm/Hz, respectively. The data rate of each user k is randomly selected from $[0.2\ 2\ 20]$ Mbps and zipf parameter α is 0.9.

We evaluate the performances of our algorithm from the following aspects: the number of BSs N, the number of users K, the number of files F and the cache size CAP. Fig. 3 indicates that the average delay decreases as the increasing of the number of BSs, where $K = 200, F = 50, CAP = 3$. It can be explained as follows: When more BSs are deployed, more files requested by users can be found in the caches at the BSs. Consequently, backhauling is not required and the total delay of can be reduced. The delay of the MC varies in a narrow range because the MC scheme can not exploit the cache diversity gain. As a result, the increasing of BSs can not reduce the delay significantly. Our proposed algorithm yields the lowest average delay as compared to others.

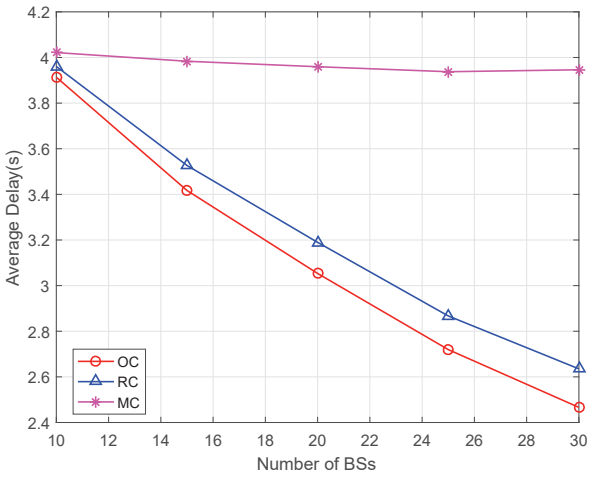

Fig. 3. Average delay as a function of the number of BSs

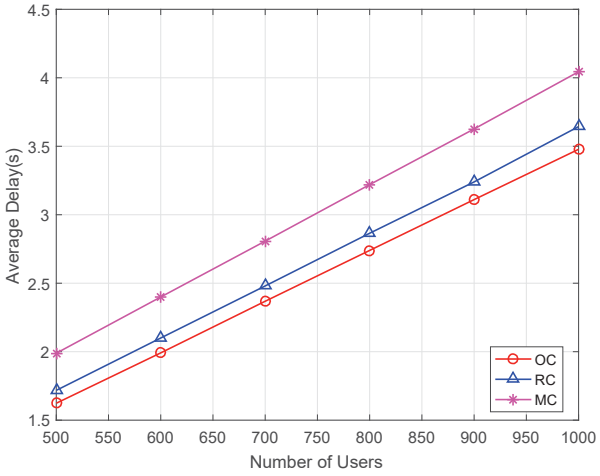

Fig. 4. Average delay as a function of the number of users

Fig. 4 shows the average delay as the function of the number of users, where $N = 20, F = 50, CAP = 3$. As can be seen from Fig. 4, as the number of users grows, the delay cost also increases proportionally. It can be explained as follows: As the number of users increases, more and more users cannot find their requested file in the local BS since the number of users served by a BS is limited due to the radio resource.

In Fig. 5, we can see the average delay increases when the number of files increases, where $K = 200, N = 20, CAP = 3$. This can be explained as follows: Given cache size of each BS, as the total number of files increases, the proportion of the cached files to the total files decreases. As a result, the probability of the requested file found locally decreases. In other words, more files should only be downloaded through the core network, which inevitably increases the average delay.

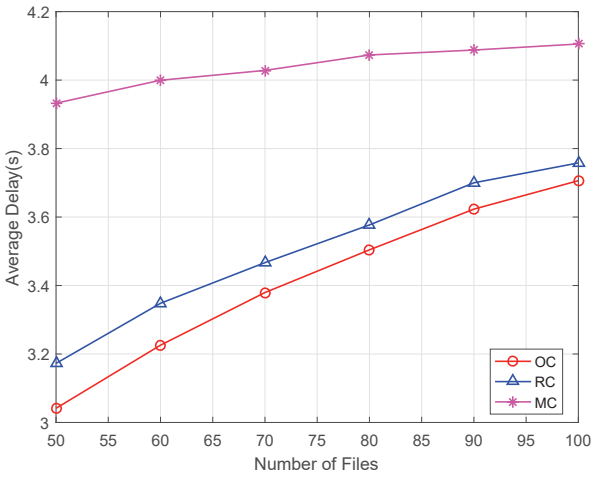

Fig. 5. Average delay as a function of the number of files

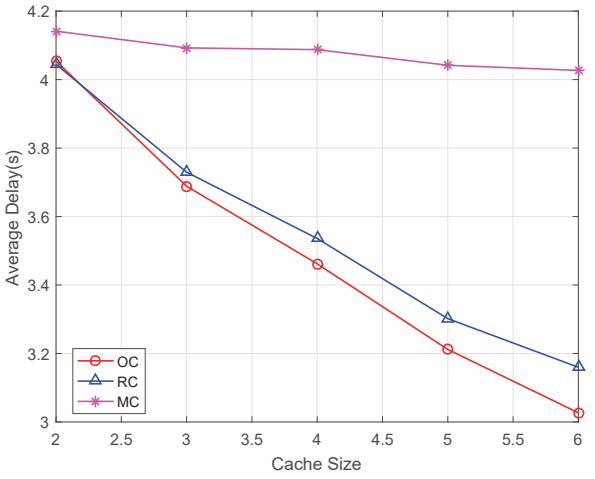

Fig. 6. Average delay as a function of cache size

Again, the MC policy yields the larger delay and is not sensitive to the number of files.

Fig. 6 illustrates how the cache size affects the average delay of all users, where $K = 200, N = 20, F = 50$. As the cache size gets larger, the average delay declines because more files can be stored in the local BS, which can improve the cache hit probability. Consequently, the average delay can be reduced. For the MC policy, since the each BS always stores the most popular files, the file diversity of the locally stored changes slightly even though the cache size increases. Our proposed algorithm yields the lowest delay as can be seen from Fig. 6.

V. Conclusions

In this paper, we study the cache placement and user association problem in cache-enabled mobile networks. Our goal is to minimize the average delay for all users in the LTE mobile communication systems. We introduce the maximum clique strategy of graph theory to solve the cache placement problem. For BSs in the same group, we place different files in each maximum clique to improve cache hit probability. Then we propose an efficient user association algorithm to serve as many as possible users while considering cache hit probability. Simulation results show that our proposed algorithm is better than other representative ones.

References

[1] C. Ran, S. Wang, and C. Wang, "Balancing backhaul load in heterogeneous cloud radio access networks," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 42–48, Jun. 2015.

[2] Q. Shen, Z. Ma, and S. Wang, "Deploying C-RAN in cellular radio networks: An efficient way to meet future traffic demands," *IEEE Trans. Veh. Techn.*, vol. 67, no. 8, pp. 7887–7891, Aug. 2018.

[3] Z. Hu, Z. Zheng, T. Wang, L. Song, and X. Li, "Caching as a service: Small-cell caching mechanism design for service providers," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6992–7004, Oct. 2016.

[4] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3520–3535, Jun. 2017.

[5] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," *IEEE J. Sel.Areas in Commun.*, vol. 35, no. 5, pp. 1076–1089, May 2017.

[6] X. Li, X. Wang, K. Li, and V. C. M. Leung, "Caas: Caching as a service for 5g networks," *IEEE Access*, vol. 5, pp. 5982–5993, 2017.

[7] J. Song, H. Song, and W. Choi, "Optimal content placement for wireless femto-caching network," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4433–4444, Jul. 2017.

[8] J. Wen, K. Huang, S. Yang, and V. O. K. Li, "Cache-enabled heterogeneous cellular networks: Optimal tier-level content placement," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5939–5952, Sep. 2017.

[9] B. Chen, C. Yang, and A. F. Molisch, "Cache-enabled device-to-device communications: Offloading gain and energy cost," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4519–4536, Jul. 2017.

[10] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas, "Video delivery over heterogeneous cellular networks: Optimizing cost and performance," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 1078–1086.

[11] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "Match to cache: Joint user association and backhaul allocation in cache-aware small cell networks," in *Proc. IEEE ICC*, Jun. 2015.

[12] Y. Wang, X. Tao, X. Zhang, and G. Mao, "Joint caching placement and user association for minimizing user download delay," *IEEE Access*, vol. 4, pp. 8625–8633, 2016.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[14] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *Proc. IEEE ICC*, London, UK, Jun. 2015.

[15] S. Wang and Y. Sun, "Enhancing performance of heterogeneous cloud radio access networks with efficient user association," in *Proc. IEEE ICC'17*, Paris, France, May 2017.

[16] X. Lin and S. Wang, "Joint user association and base station switching on/off for green heterogeneous cellular networks," in *Proc. IEEE ICC'17*, Paris, France, May 2017.