A GA-based Mobility-aware Proactive Caching Strategy in Heterogeneous Ultra-Dense Networks

Ning Gao, Xiaodong Xu, Lei Gao
National Engineering Lab for Mobile Network Technologies
Beijing University of Posts and Telecommunications,Beijing,100876,China
Beijing, China
Email: gaoning@bupt.edu.cn

Abstract—Caching on the wireless edge is a promising solution to tackle the backhaul constraint of network densification and reduce the delay of content transmission. Although some effective caching strategies have been introduced to cellular networks, most of them ignored user mobility, which is unreasonable. In reality, users are mobile and the association with small base stations (SBSs) can change during the file downloading, which can be more frequent in heterogeneous ultra-dense networks (HetUDNs). Cache miss in another SBS will greatly increase the delay of file transmission. In this paper, we propose a mobility-aware proactive caching strategy in HetUDNs, which can proactively schedule content into different cache layer to improve users Quality of Service (QoS) and cache hit ratio. The cache placement problem is formulated with the aim of effective capacity maximization and solved by a genetic algorithm (GA)-based approach. Simulation results show that the proposed scheme performs better than the most popular caching (MPC) strategy.

Index Terms—Mobility, Caching Strategy, Heterogeneous ultra-dense networks (HetUDNs), Effective Capacity, Genetic Algorithm

I. INTRODUCTION

With the development of mobile Internet, a large number of new applications are widely applied, such as AR/VR and online games. These applications require the mobile network provides the higher data transmission rate and lower network delay. Moreover, it is predicted that the mobile data traffic will reach 30.6 exabytes/month globally by 2020 [1]. To handle this challenge, heterogeneous ultra-dense networks (HetUDNs) with dense small base stations (SBS) and macro base station (MBS) has been widely introduced to enhance the network capacity. However, the costly and heavily-loaded backhaul link between HetUDNs and the core network is becoming the bottleneck [2].

Caching in the HetUDNs has been widely recognized as an effective and economical solution to tackle the aforementioned challenges. By storing popular files in the densely deployed SBSs in advance [3], more requests can be satisfied at SBSs instead of retrieving duplicated file over the backhaul links. Besides, downloading directly from SBSs can reduces delay substantially due to the short transmission distance [4]. However, the storage capacity on SBSs is usually limited, and thus caching strategies should be carefully designed to make the best use of the SBSs cache.

There have been some recent works focusing on designing caching strategies in the scenario of HetUDNs. The auther of [5] present a wireless distributed caching strategy with a low-bandwidth backhaul link but high storage capacity where users can access multiple SBSs. Finding the optimal cache placement to maximize cache hit ratio is proved to be NP-complete. The authors of [6] study the optimization issue for cache content placement in caching enabled HetUDNs with heterogeneous file and cache sizes and adopt multicast transmission to minimize the average backhaul rate. In the work of [7], the authors propose a cooperative caching strategy where SBSs can get the desired content from neighboring SBSs so as to enhance the content delivery efficiency.

Although these recent works have provided insight into caching strategy cellular networks, users are assumed to be static and the association between users and SBSs remains unchanged. Obviously, such an assumption is unreasonable in mobile networks. Users are moving and the association to SBSs can change during the file downloading, which can be more frequently in HetUDNs [8]. The cache placement policy that ignores user mobility cannot capture the spatial distribution changes of the file request in time, resulting in a lower degree of matching between the cache and the request. Therefore, the impact of user mobility cannot be neglect when designing file caching strategies [9]. Reference [10] uses a discrete random jump model to describe the mobility pattern and derive the expression of throughput. Due to the complexity of the problem, two heuristic algorithms are provided to obtain the optimal solution. The author of [11] assumes that the user movement obeys a discrete Markov model and the amount of data downloaded by users depends on their location and the cache placement in each time slot. The user mobility is also modeled as Markov chain and a distributed caching strategy is proposed in a two-tier heterogeneous network with the aim of minimizing the content fetched from macro base station (MBS) [12].

Although these works have taken user mobility into account, they have not fully utilized the hierarchical architecture of heterogeneous networks to address the problems caused by user mobility. In a HetUDN, both MBS and SBS can be equipped with cache devices, but the delay in obtaining content from them is different. The SBS is close to the user with a

small delay while the MBS is relatively far with a slightly larger delay. And the requests that the SBS miss are redirected to the MBS [12]. Assuming that the user only accesses SBSs, the high-speed movement will cause the user to frequently switch between multiple SBSs. Due to the randomness of user movement, it is very difficult to push the required files to SBSs accurately in advance to reduce the delay, and the cache hit ratio is hard to guarantee. On the contrary, if these contents are more likely to be cached in the MBS cache, the cache hit rate of high-speed users can be effectively improved at the cost of a slightly larger delay. In addition, the SBSs cache can store more other content to satisfy more low-speed users' content requests.

In this paper, we are motivated to propose a mobility-aware proactive caching strategy to improve users QoS and cache hit ratio. Content can be scheduled in different cache layer depending on user mobility. The contributions of this paper include:

- We propose a mobility-aware proactive caching strategy in HetUDNs to provide the moving user with better QoS.
- The cache placement problem is formulated as a 0-1 integer nonlinear programming problem with the aim of effective capacity maximization, which can reflect the impact of delay in user data rate.
- We solve the problem by a genetic algorithm (GA)-based approach. Simulation results show that the proposed strategy can achieve better performance compared with the most popupar caching (MPC) strategy.

The remainder of this paper is organized as follows. Section II describes the system model, including network model, file popularity model, mobility model and effective capacity based QoS parameter. In Section III, we introduce the proposed mobility-aware proactive caching strategy and solve the cache placement problem with the GA-based approach. In Section IV, numerical results are illustrated with performance comparisons between MPC. Finally, conclusions are summarized in Section V.

II. SYSTEM MODEL

A. Network Model

In this paper, we consider a heterogeneous network comprising a single MBS and K densely deployed SBS denoted as $\mathbf{S} \triangleq \{0, 1, 2, ..., K\}$, where S_0 represents the MBS. The MBS and SBSs are connected by a fronthaul link with the limited capacity of r_{FH} and the MBS is connected to the core network by a backhaul link with the limited capacity of r_{BH}. The set of active users is $\mathbf{U} \triangleq \{1, 2, 3, ..., U\}$ and each user only requests one file at a time while moving from one cell to another from time to time. We consider that each user u only connects to and receives data from the nearst BS (in terms of signal strength), which we later refer to as the user *us* home BS. And the users set associate with SBS S_k is denoted as U_k.

We assume that the MBS has access to a library of F files denoted as $\mathbf{F} \triangleq \{1, 2, 3..., N\}$, and each file has the same
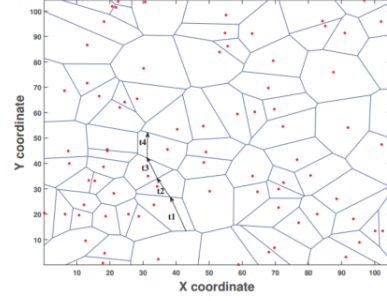

Fig. 1. System Model.

size denoted by B[MB]. Note that the assumption of equal size files is justifiable in practice since files can be divided into blocks of the same size.

We assume that SBS i are equipped whih a cache to storage files with storage capacity of $C_i[MB]$, and the MBS is equipped with a large cache with storage capacity of $C_0[MB]$. Each user requesting for files in $\mathbf{F}$ is initially served by the SBS he contacts. If the requested file is not present in the cache, other SBSs nearby or the MBS will send the file to the user as shown in Fig.1. To describe the cache placement decision, we define the cache placement matrix as:

$$\mathbf{X} = \begin{Bmatrix} x_{0,1} & x_{0,2} & \cdots & x_{0,N} \\ x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{K,1} & x_{K,2} & \cdots & x_{K,N} \end{Bmatrix}$$

where $x_{i,j}$ denotes caching the file f_j in BS i. Note that the indexing i of $x_{i,j}, i = 0, 1, ..., K$, includes all the MBS and SBSs. The cache storage capacity constraints can be expressed as:

$$\sum_{j=0}^{F} B \cdot x_{i,j} \quad < C_i, \forall i \in \mathbf{S} \tag{1}$$

B. File Popularity Model

We define the populary distribution of files in cell i as $\mathbf{P} \triangleq \{p_{i,1}, p_{i,2}, p_{i,3}..., p_{i,F}\}$, in which $p_{i,j}$ is the probability of the file f_j being requested from a user in BS i, and $\sum_{j=1}^{N} p_{i,j} = 1, \forall i \in \mathbf{S}$.It should be noted that the content populary here is inferred with historical data and may not reflect the file request probability in the next time period due to user mobility.

C. Mobility Model

User mobility is modeled by discrete random jumps, with the corresponding intensity characterized by average sojourn time. In terms of the distribution of sojourn time, it is reasonable to model it by an exponential function [10]. Therefore, the probability density function (PDF) of sojourn time in BS i can be written as follows:

$$p(\tau, \mu) = \frac{1}{\mu} e^{-\tau/\mu} \tag{2}$$

where μ is the average sojourn time. A small μ_i indicates intense mobility and vice versa.

In addition, in order to predict the trajectory of moving users, we adopt the stationary Markov chain model. Let $T_{l,i} \geq 0$ denote the probability of transition from cell l to cell i, where $\sum_{i \in \mathbf{S}} T_{l,i} = 1, for all$. While predicting the cell transition probability is a challenging task in terms of accuracy, the recent advances in machine learning techniques have made great progress on achieving this goal.

D. Effective Capacity based QoS parameter

Although introducing caching technology into RAN shortens the access distance between users and contents significantly, which can improve the service quality and efficiency of the network, it is not easy to find a proper measure to evaluate this advantage. Conventional channel models are formed from the perspective of physical layer, so it is difficult to evaluate the QoS supporting ability of the channel, such as bounds on delay and packet loss ratio. In [13], the authors construct the channel model from the link layer and analyze the behavior of the connection under complex QoS requirements with the queuing theory. The maximum arrival rate the wireless channel can supported is defined as a log-moment generation function as follow

$$\alpha(\theta) = -\lim_{t \to \infty} \frac{1}{\theta t} \log E e^{-\theta S(t)} \tag{3}$$

where $S(t) = \int_0^t r(t)\, dt$ represents the data throughput accumulated on the time domain. Considering the scenario that channel coefficients keep constant over the frame duration T and vary independently for each frame, the formula of effective capacity can be rewritten as follow:

$$\alpha(\theta) = -\lim_{t \to \infty} \frac{1}{\theta t} \log E e^{-\theta TR[i]} \tag{4}$$

where $R[i]$ indicates the instantaneous channel capacity during the ith frame.

Due to the time-varying and randomness of wireless channels, deterministic delay guarantees for specific links are unreasonable and impractical. So, the effective capacity denotes θ as the statistical QoS guarantee parameter and the delay violation possibility can be written as

$$Pr\{D(\infty) > D_{max}\} \approx e^{-\theta D_{max}} \tag{5}$$

where D_{max} indicates the delay bound and a larger θ represents a better link quality or a tighter QoS constraints. Note that when θ approaches to zero, the effective capacity converges to the ergodic capability.

According to the queuing theory, the queue delay $D(\infty)$ of the packet satisfies the following relation [31].

$$\theta = -\lim_{D_{max} \to \infty} \frac{\log(Pr\{D(\infty) > D_{max}\})}{D_{max} - \sum_i d_i} \tag{6}$$

where d_i is the transmission delay caused by the i-hop in data packet ransmission.

In the scenario of this paper, the transmission delay can be different accroding to the cache placement as shown in Fig.2
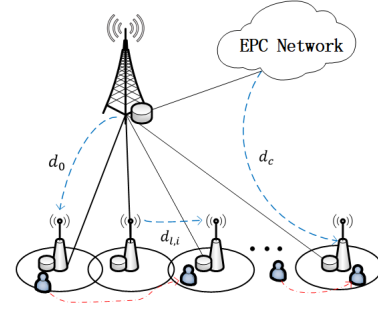

Fig. 2. Delay Model.

Home SBS cache hits do not need transmission delay. The transmission delay of MBS transfering file to home SBS is $d_0 = \frac{B}{r_{FH}}$.

Sharing files between SBS l and home SBS i will cause different transmission delay depending on the hops number h_l, i, which can be denoted by $d_{l,i} = h\frac{B}{r_{FH}}$.

Retrieving files from the EPC network through the backhaul link generally results in a large transmission delay $d_C = h_C \frac{B}{r_{BH}}$, where h_C is a large constant value.

The place where user can get the requested file affects the effective capacity of the wireless link, as the value of the QoS guarantee factor changes according to $x_{i,j}$. Thus given a cache placement matrix $\mathbf{X}$, we can get the average transmission delay of user u in cell i requesting file j as,

$$D_{u,i,j} = (1 - x_{i,j})(x_{o,j}d_0 + \frac{1 - x_{o,j}}{K - 2} \sum_{l \in S \setminus \{o,i\}} x_{l,j}d_{l,i}) + c_{i,j}d_c \tag{7}$$

where $c_{i,j} = \prod_{l \in S} 1 - x_{i,j}$, which means file j is not storaged in RAN.

Substituting $D_{u,i,j}$ to equation (6), the QoS guarantee parameter can be expressed as,

$$\theta_{u,i,j} = -\lim_{D_{max} \to \infty} \frac{\log(Pr\{D(\infty) > D_{max}\})}{D_{max} - \sum_i d_i} \tag{8}$$

III. GA-BASED MOBILITY-AWARE PROACTIVE CACHING STRATEGY

Here, we first explore the impact of user mobility on file popularity of different base stations. Then, we cast the cache placement optimization problem, which is aimed at maximizing the system effective capacity.

A. Mobility-aware content request probability

When a user with an incomplete download session of content j arrives at cell k, part of the file, $B_{k,j}$, has been downloaded already. Hence the user only needs to download the rest of the content j. For simplicity, we assume the maximum bandwidth of G [MB/s] of each SBS is shared equally among the associated users of that SBS. Thus, the bandwidth allocated to each user connecting to BS l is given by $r_l = \frac{G}{U_l}$.

Since each content file has B MB, the time needed to completely download one content at cell 1 is $t_l = B/r_l$.

However, the sojourn time that user stays in cell l is subject to exponential distribution, so the probability of the user completing the download when leaving the cell l can be inferred using the Cumulative Distribution Function $F(\tau, \mu_l)$ as,

$$
\begin{aligned}
q_{l,j} &= Pr(\tau < t_l - \frac{B_{l,j}}{r_l}) \\
&= F(t_l - \frac{B_l, j}{r_l}) \\
&= 1 - exp(-(t_l - \frac{B_{l,j}}{r_l}/\mu))
\end{aligned}
\tag{9}
$$

Then,the file requested probability $p_{(\tilde{i},j)}$ for content j at cell i is,

$$
p_{\tilde{i},j} = \begin{cases} p_{i,j} + \lambda \sum_{l \neq i} p_{l,j} q_{l,j}, & i = 0 \\ p_{i,j} + (1-\lambda) \sum_{l \neq i} p_{l,j} q_{l,j} T_{l,j}, & i \in S \setminus 0 \end{cases}
\tag{10}
$$

Where $p_{i,j}$ is the prior popularity of file j at cell i and $T_{(l,i)}$ is the transition probability from cell l to i. And $\lambda \in (0,1)$ is cosntant determined empirically, controlling the tendency of the file cache hierarchy. A larger λ factor will make files with higher mobile strength tend to be cached in the MBS cache, voiding cache misses caused by frequent handovers among SBS. Conversely, a smaller λ factor will make the file tend to be cached in the SBS cache closer to the user, minimizing the transmission delay.

B. Problem Formulation

Based on the system model and effective capacity theory previously mentioned, the effective capacity achieved by user u in cell i can be expressed by the following formulation:

$$
\alpha_{u,i}(\theta) = \sum_j p_{\tilde{i},j}(-\frac{1}{\theta_{u,i,j}T} \log E e^{-\theta_{u,i,j} T r_i})
\tag{11}
$$

The problem can be formulated as follows with the purpose of maximizing the sum effective capacity of the whole system.

$$
\begin{aligned}
max \quad & \sum_u \sum_i \alpha_{u,i}(\theta) \\
s.t. \quad & \sum B \cdot x_{i,j} < C_i \\
& x_{i,j} \in \{0,1\}
\end{aligned}
\tag{12}
$$

Obviously, the optimal problem is a 0-1 integer nonlinear programming problem, and it is highly complicated to obtain a close form solution. In order to solve the problem, the genetic algorithm is adopted to solve this problem.

C. The GA-based approach

The Genetic algorithm is an adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection, and it is inherently suitable for solving optimization problems with binary variables [14].

Firstly, N_p candidate caching placement matrices are generated, known as the initial population, and each matrix is called an individual. Then the objective value of each individual is calculated through equation (11). N_e individuals with top objective values are chosen as elites and selected into the next generation directly. The rest of the next generation population are generated through crossover and mutation operations. The crossover function selects two individuals parent individuals to generate two crossover children, and the mutation function just operates on a single individual and generates a mutation child. The numbers of individuals generated through crossover and mutation operations are denoted as N_c and N_m , respectively, where $N_e + N_c + N_m = N_p$ and the crossover fraction is $f_c = N_c \setminus (N_c + N_m)$. Roulette wheel selection is adopted, and individuals with higher objective values in current generations will have a higher probability to generate offsprings. Repeat the evaluation-selection-generation procedures until Ng generation is created. Finally, the best individual in the current population is chosen as the output of the algorithm. The initial population, crossover function and mutation function of the proposed GA approach are described as follows.

1) Initial Population: The initial population is created as a set of $\mathbf{X}^{(K+1)F}$. For each row in each individual, M_i of the F elements are set to be one randomly, and all the left elements are set to zero, where

$$
\sum_{j=0}^{K} B \cdot M_i \quad < C_i, \forall i \in \mathbf{S}
\tag{13}
$$

for the storage capacity constraint in each BS.

2) Crossover Function: The crossover function use a two-point crossover function to generate one children, which is described in Algorithm 1. The steps 9-14 are heuristic processes to meet constraint (13).

3) Mutation Function: The mutation function operates on a single individual and generates its mutation child. For each row of the individual, one of F elements is randomly selected and the value is set to be the opposite. The mutation operation reduces the probability that the algorithm converges to local minimums.

The complexity of the proposed GA is $N_p * N_g$, where N_p and N_g are the population size and the number of generations evaluated, respectively.

IV. PERFORMANCE EVALUATION

In this section, we use computer simulation methods to evaluate the performance of the mobility-aware proactive caching strategy for HetUDNs. We simulated a HetUDN depicted in Fig1, where an MBS is located in the center and multiple SBSs are uniformly deployed in the network. Several moving users are randomly distributed in the network and we consider a wrap-around network layout such that when a user moves out of the network on one side, it comes back in on the opposite side w.r.t. the origin. We assume that the files users request can be modeled as the Zipf distribution.The request probability of file f_j is

$$
p_j = \frac{1/j^\gamma}{\sum_{i=1}^{N} 1/j^\gamma}, \quad \forall j \in F
$$

Algorithm 1 Crossover function

1: Get parent $\mathbf{X}_1 = \{x_{i,j}^{(1)}\}$ and $\mathbf{X}_2 = \{X_{i,j}^{(2)}\}$ from selection function, initialize their child $\mathbf{X}_c = \{x_{i,j}^{(c)}\} = \mathbf{0}^{(K+1)\times F}$

2: **for** $j = 1, 2, \cdots, F$ **do**

3:    Generate random integers $l_1, l_2 \in [1, K+1], l_1 \neq l_2$

4:    **if** $l_1 < l_2$ **then**

5:       Replace $x_{i,j}^{(1)}, l = \{l_1, l_1 + 1, \cdots, l_2$ of $\mathbf{X_1}$ with $x_{i,j}^{(2)}, l = \{l_1, l_1 + 1, \cdots, l_2$ of $\mathbf{X_2}$, and then set $x_{i,j}^{(c)} = x_{i,j}^{(1)}, \forall l \in \{1, 2, \cdots, L\}$.

6:    **else**

7:       Replace $x_{i,j}^{(2)}, l = \{l_2, l_2 + 1, \cdots, l_2$ of $\mathbf{X_2}$ with $x_{i,j}^{(2)}, l = \{l_2, l_2 + 1, \cdots, l_2$ of $\mathbf{X_1}$, and then set $x_{i,j}^{(c)} = x_{i,j}^{(2)}, \forall l \in \{1, 2, \cdots, L\}$.

8:    **end if**

9:    **while** $\sum_{l=1}^{L} x_{i,j}^{(c)} > M_n$ **do**

10:      Set nonzero $x_{i,j}^{(c)}$ to 0 in descending order of l.

11:    **end while**

12:    **while** $\sum_{l=1}^{L} x_{i,j}^{(c)} < M_n$ **do**

13:      Set nonzero $x_{i,j}^{(c)}$ to 1 in ascending order of l.

14:    **end while**

15: **end for**

where γ which is the skewness reflecting the concentration of the popularity distribution among users. All other simulation parameters are listed in Tabel 1.

TABLE I
SIMULATION PARAMETERS

System Parameter	
Number of MBS	1
Number of SBSs K	100
Fronthaul rate r_{FH}	600Mbps
Backhaul rate r_{BH}	1000Mbps
Bandwidth of SBS G	100Mbps
Number of Files N	200
Size of Files B	20MB
GA Parameter	
Population size N_p	50
Terminal generation number N_g	100
Number of elites N_e	10
Crossover fractio f_c	0.8

We compare the performance of our optimal caching scheme with MPC (most popular caching) schemes, which uses global static content popularity and each BS stores the most popular files until its cache is full [5].

A. Capacity Based System Tthroughput Performance

Fig.3 shows the effective capacity based system throughput when the num of users varies in the network, where we set $\gamma = 0.7$ and $\mu = 4$. Obviously, we see that as the number of user increases, the system throughout improves and the mobility-aware proactive caching strategy always outperforms the MPC strategy. This is because that files cached in the SBS and MBS changes according to users location and preference. Hence, it

can match the user request better and reduce the transmission delay.
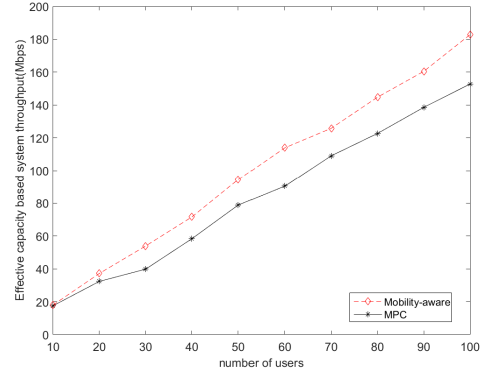

Fig. 3. System throughput gain with the proposed caching strategy

Fig.4 investigate the relationship between cache size and system throughput. As shown in Fig.4, with the total cache size increases, the effective capacity based system throughput grow at the same time. It demonstrates the fact that expense of storage can alleviate the shortage of backhaul bandwidth and improve the data rate that a user achieves. The more contents BSs cache, the more chances users can get the contents in RAN so that they can be served immediately.
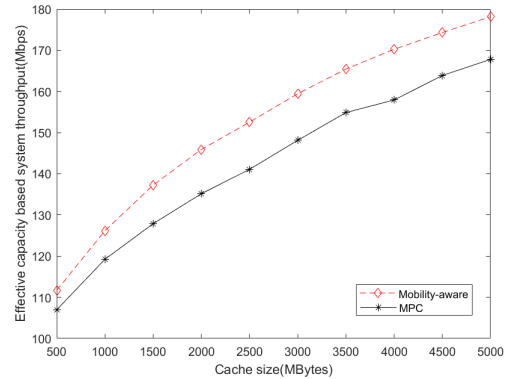

Fig. 4. System throughput versus cache capacity

B. Cache Hit Ratio Performance

Fig.5 display the impact of file populality distribution parameter γ on the cache hit ratio. In this simulation, user number is set to 70. As the figure shows, when γ small, both two caching strategy have low cache hit ratio. Because in this case, all files are nearly equally likely to be requested and the storage capacity of BS is limited. But mobility-aware caching strategy still achieves superior performance than MPC. When γ becomes larger, the requests are more concentrated among some popular files. Therefore, these files are more likely to be cached in most of the SBS resulting in hitting more user requests. Hence the caching hit ratio of mobility-aware and MPC both improve notably.
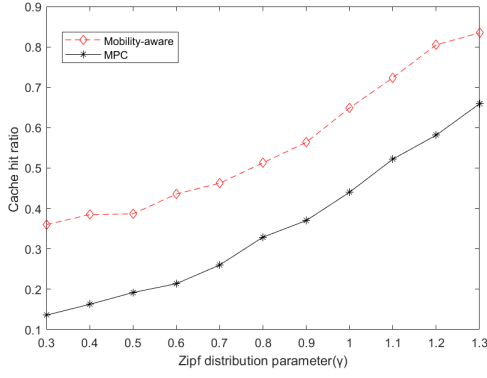
Fig. 5. Cache hit ratio as a function of Zipf distribution parameter γ

V. CONCLUSION

In this paper, we proposed a new mobility-aware proactive caching strategy for heterogeneous ultra-dense network and use the effective capacity to evaluate the effect of transmission delay on date rate. We use random jump model and stationary Markov model to describe the mobile pattern of users and predict the popularity at BSs with it. Then, we formulated the optimal content placements problem as a 0-1 integer nonlinear programming problem and solved it by genetic algorithm based approach. Finally, simulation results show that the proposed mobility-aware proactive caching strategy achieves higher throughput and cache hit ratio than MPC strategy while users are moving.

REFERENCES

[1] C. V. Mobile, "Cisco visual networking index: Global mobile data traffic forecast update, 2011-2016," 2014.
[2] X. Ge, H. Cheng, M. Guizani, and T. Han, "5g wireless backhaul networks: challenges and research advances," *IEEE Network*, vol. 28, no. 6, pp. 6–11, Nov 2014.
[3] Z. Ye, C. Pan, H. Zhu, and J. Wang, "Tradeoff caching strategy of the outage probability and fronthaul usage in a cloud-ran," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6383–6397, July 2018.
[4] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5g systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, February 2014.
[5] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec 2013.
[6] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Multicast-aware caching for small cell networks," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014, pp. 2300–2305.
[7] Q. Jia, R. Xie, T. Huang, J. Liu, and Y. Liu, "Energy-efficient cooperative coded caching for heterogeneous small cell networks," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, May 2017, pp. 468–473.
[8] Z. Zhou, M. Dong, K. Ota, and Z. Chang, "Energy-efficient context-aware matching for resource allocation in ultra-dense small cells," *IEEE Access*, vol. 3, pp. 1849–1860, 2015.
[9] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: modeling and methodology," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 77–83, August 2016.
[10] X. Liu, J. Zhang, X. Zhang, and W. Wang, "Mobility-aware coded probabilistic caching scheme for mec-enabled small cell networks," *IEEE Access*, vol. 5, pp. 17 824–17 833, 2017.
[11] K. Poularakis and L. Tassiulas, "Exploiting user mobility for wireless content delivery," in *2013 IEEE International Symposium on Information Theory*, July 2013, pp. 1017–1021.
[12] ——, "Code, cache and deliver on the move: A novel caching paradigm in hyper-dense small-cell networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 675–687, March 2017.
[13] D. Wu and R. Negi, "Effective capacity: a wireless link model for support of quality of service," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 630–643, July 2003.
[14] M. Srinivas and L. M. Patnaik, "Genetic algorithms: a survey," *Computer*, vol. 27, no. 6, pp. 17–26, 2002.