**GA-based**

**I. Introduction**

With the development of mobile Internet, a large number of new network services and applications are widely applied, such as AR/VR, mobile high definition video, online game etc., which require that the mobile network provides the higher data transmission rate and lower network delay. Moreover, it is predicted that by year 2020, the mobile data traffic will reach 30.6 exabytes/month globally [1]. To handle the severe challenge, densely deployed small cell network (SCN) has been widely introduced by network operators to enhance the network capacity. However, the costly and heavily-loaded backhaul link between SCN and the core network is becoming the bottleneck.

Caching on the SCN has been widely recognized as an effective and eonomical solution to tackle the aforementioned problem of network densification. By storing popular files in the SBS cache in advance [2][3], more requests can be satisfied at SBSs instead of retrieving duplicated file over the backhaul links. Besides, downloading directly from SBSs reduces download delay substantially due to the short transmission distance and thus high rate. However, the storage capacity on SBSs is generally limited, and thus caching strategies should be carefully designed to make the best use of the SBS cache.

There have been some recent works focusing on designing file caching strategies in the scenario of network densification. Reference [FemtoCaching] present a wireless distributed caching stratrgy with a low-bandwidth backhaul link but high storagr capacity where user can access multiple SBSs. Finding the optimal cache placement to maximize cache hit ratio is proved to be NP complete. The authors of [cluster hetnet coded] study the optimization issue for cache content placement in caching enabled SCN with heterogeneous file and cache sizes, and adopt multicast transmission to minimize the average backhaul rate. In [Energy-efficient Cooperative Coded Caching for Heterogeneous Small Cell Networks], the authors propose an cooperative caching strategy where SBS can get the desired content from neighbouring SBSs so as to enhance the content delivery efficiency.

Although these recent works have provided insight into caching strategy in cellular network, users are assumed to be static and the association between users and SBSs remains unchanged. Obviously, such assumption is unreasonable. Users are mobile and the association to SBSs can change during the download of a file, which can be more frequently with densely deployed SBSs. The cache placement policy that ignores user mobility cannot capture the spatial distribution change 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 can not be neglect when designing file caching strategies [Mobility-aware caching for content-centric wireless networks: modeling and methodology]. Reference [Exploiting user mobility for wireless content delivery] 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 author of [Mobility-Aware Coded Probabilistic Caching Scheme for MEC-Enabled Small Cell Networks] uses a descrete 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. Reference [Code, cache and deliver on the move: a novel caching paradigm in hyper-dense small-cell networks] model user mobility as Markov chain and put for forward a distributed caching paradigm in a two-tier heterogeneous network with the aim of minimizing the content fetched from MBS.

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 an ultra-dense heterogeneous network, both MBS and SBS can deploy cache devices, but the delay in obtaining content from them are different. The SBS is close to the user with small delay while the MBS is relatively far with a larger delay, and the request that the SBS misses is redirected to the MBS. Assuming that the user only accesses SBSs, high speed movement will cause the user to frequently switch between multiple SBSs. Due to the randomness of user motion, it is very difficult to practively push the required files to SBSs accurately in advance to reduce the delay, and the cache hit rate 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 a certain delay. In addition, the SBS 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 coordinate 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:

1) We analysis the effect of mobility on file popularity at different base station, and propose a mobility-aware proactive layered caching strategy.

2) The effective capacity is adopted to evaluate the system performance, which can reflect the impact of delay in user date rate.

3)We formulate the cache placement problem as a 0-1 integer nonlinear programming problem, and solve the problem by a genetic algorithm(GA)-based approach. Extensive simulation results show that the proposed strategy can achieve better performance compared with the existing works.

The remainder of this paper is organized as follows. Section II describes the system model, including network model, file popularity model, mobility model and transmission delay model. In Section III, we introduce the proposed mobility-aware coordinate proavite caching strategy and fromulate the cache placement problem as an optimization problem. In Section IV, a genetic algorithm is adopted to solve the problem. In Section V, numerical results are illustrated with performance comparisons between MPC. Finally, conclusions are summarized in Section VI

**II. System Model**

1. *Network model*

In this paper, we consider a heterogeneous network comprising a single MBS and K densely deployed SBS. All the BSs are denoted as where represents the MBS. The MBS and SBSs are connected by a fronthaul link with limited capacity of and the MBS is connected to the core network by a backhaul link with limited capacity of . The set of active users is and each user only requests one file at a time while moving from one SBS to another from time to time. We consider that each user only connects to and receives data from the nearst SBS (in terms of signal strength), which we later refer to as the user home SBS. And the users set associate with SBS i is denoted as .

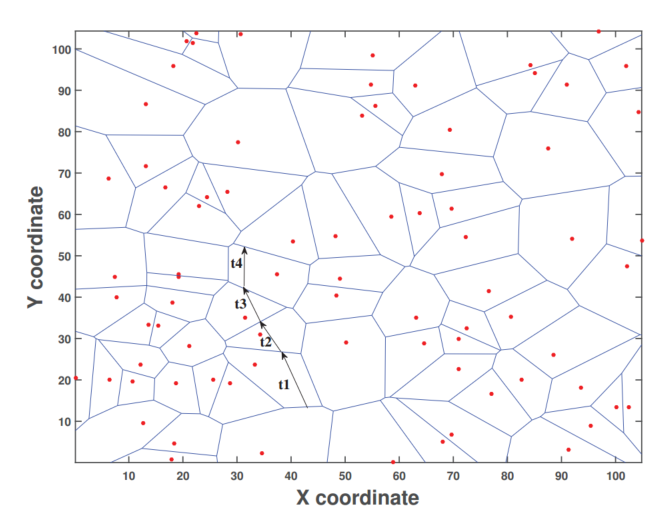


Fig.1(暂代)

We assume that the MBS has access to a library of files denoted as , and each file has the same size denoted by [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 [MB], and the MBS is equipped with a large cache with storage capacity of [MB]. Each user requesting for files in *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:

where denotes caching the file in BS i. Note that the indexing i of ,i=0,1,…,K, includes all the MBS and SBS. The cache storage capacity constraints can be expressed as:

1. *File Popularity Model*

We define the populary distribution of files in BS as , in which is the probability of the file being requested from a user in BS , and . It should be noted that the content populary here is inferred with historical data and can not reflect the short-term variation due to user mobility.

1. *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 [MEC]. Therefore, the probability density function (PDF) of sojourn time in SBS can be written as follows:

where is the average sojourn time. A small 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 denote the probability of transition from SBS to SBS , where . 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. When a user has incomplete download at one SBS and enters the new SBS, the content is requested at the new SBS thus affecting the static content popularity .Therefore, the popularityof contentin BS may not reflect the actual request probability of content in BS . If the content requested by user is not cached in the new SBS, it has to be fetched from core network via high delay link. In order to provide a better QoS service, we need to dynamically adjust the distribution of file popularity based on the user's mobile pattern, thereby reducing the huge latency caused by cache misses.

1. *Transmission delay Model*

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 [8], 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

where 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:

where indicates the instantaneous channel capacity during the i-th 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 as the statistical QoS guarantee parameter. As [9] mentions, the delay violation possibility can be written as

where indicates the delay bound and a larger represents a better link quality or a tighter QoS constraints. Note that when approaches to zero, the effective capacity converges to the ergodic capability [10].

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 . [11]. According to the queuing theory, the queue delay of the packet satisfies the following relation [31]*.*

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

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



Fig. 2

Home SBS cache hits do not need transmission delay. The transmission delay of transfering file from MBS to home SBS is:

Sharing files between SBS l and home SBS i will cause different transmission delay depending on the hops number , which can be denoted by:

Retrieving files from the core network through the backhaul link generally results in a large transmission delay [octopus]:

So, given a cache placement matrix *X*, we can get the average transmission delay of user u in SBS i requesting file j as,

where , which means file j is not storaged in RAN.

Substituting to equition (), the QoS guarantee parameter can be expressed as,

**III. Mobility-aware Caching Strategy and Problem Formulation**

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

1. *Mobility-aware Caching Strategy*

Mobility-aware content request [probability](javascript:;): When a user with an incomplete download session of content j arrives at SBS l, part of the file, 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 BS is shared equally among the associated users of that BS. Thus, the bandwidth allocated to each user connecting to BS *l* is given by

Since each content file has B [MB], the time needed to completely download one content at SBS l is .

However, the sojourn time that user stays in SBS l is subject to exponential distribution, so the probability of the user completing the download when leaving the SBS l can be inferred using the Cumulative Distribution Function as,

Then, the file requested probability for content j at BS i is,

Where is the prior popularity of file j at cell i and is the transition probability from cell l to i. And is cosntant determined empirically, controlling the tendency of the file cache hierarchy. A larger 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 factor will make the file tend to be cached in the SBS cache closer to the user, minimizing the transmission delay.

1. *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:

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

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

**IV. The Solution of Optimal Problem**

Genetic Algorithm [5] is adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GA is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. The processes of GA include coding, colony initialization, initialization, crossover and mutation.

The GA maintains a population of possible cache placement solutions. A solution corresponds to a chromosome which is an encoded representation of the cache placement of all the SBS and MBS.

1. *Coding*

We choose the linear structure coding method, in which a chromosome represents a feasible solution of the cache allocation problem. There are N(K+1) elements in each chromosome as shown in fig3; each element records the allocation information in the corresponding cache placement matrix X.

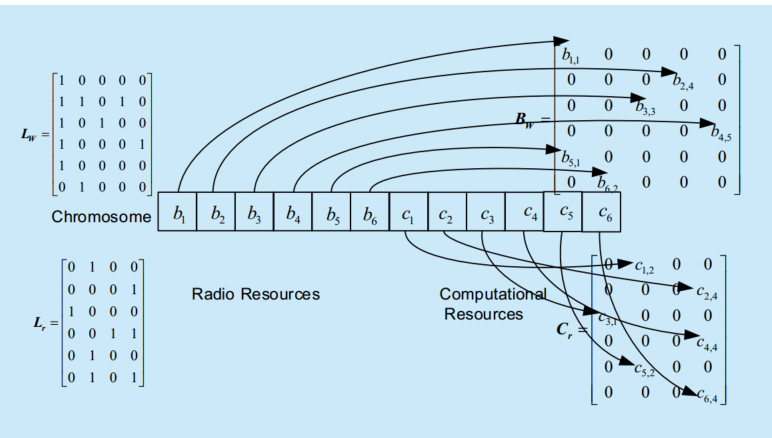


Fig.3 coding of genetic algorithm(暂代)

1. *Colony initialization*

We generate Zg chromosomes with number of 1 randomly distributed so as to satisfy the cache capacity constrain.

1. *Fitness function*

Fitness function is the system optimal function defined in equation [(1)](https://ieeexplore.ieee.org/document/#deqn1).

1. *Selection*

The chromosomes in the colony are ranked, then the first x% (Elite selection ratio) is selected and the remains are selected at random. This strategy can guarantee that the genetic algorithm converges to the global optimal solution.

1. *Crossover*

For two chromosomes to be mated which are chosen with crossover probability Pc, exchange the each corresponding element with probability Ps (crossover ratio).

1. *Mutation*

For each chromosome chosen with mutation probability Pm, replace each element with a random value with probability Pb (mutate ratio).

1. *Algorithm process*
2. Initialize colony.
3. Evaluate the individual in colony using fitness function, if the evolutionary generation reaches to the terminal generation then jump to step 4, otherwise continue.
4. Generate the new colony using selection, crossover and mutate operator, then jump to step 2.
5. Get the cache placement matrix through the best individual generated by genetic algorithm.
6. Deliver files to each BS accroding to the cache placement matrix. Cache placement in this cycle is over.

**V. Performance Evaluation**

In this section, we use computer simulation methods to evaluate the performance of the mobility-aware proactive caching placement scheme for heterogeneous ultra density networks. We simulated a heterogeneous ultra density network depicted in Fig1, where the MBS is located in the center and 16 SBSs are uniformly deployed in the network. Moreover, we devided the macrocell into 4 clusters, and each cluster consists of 4 SBSs. Additionally, 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. There several users in the network and the initial location of the users is random, following a uniform distribution. We assume that the files users request can be modeled as the Zipf distribution. The request probability of file is

,

where is the skewness reflecting the concentration of the popularity distribution among users. Moreover, we consider a library of N=200 files, each of size B=50 MB. The popularities of the files at each BS are generated randomly simulating different file preferences in different regions. The backhaul rate is set to = 1000Mbps and the fronthaul rate 600Mbps.

The paremeter of genetic algorithm are set as table 1 shown,

|  |  |
| --- | --- |
| Colony size *Zg* | 50 |
| Terminal generation number | 100 |
| Crossover probability *Pc* | 0.75 |
| Mutation probability *Pm* | 0.02 |
| Reserve ratio x％ | 25％ |
| Crossover ratio *Ps* | 0.5 |
| Mutation ratio *Pb* | 0.05 |

Tabel 1. genetic algorithm parameter

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 populay files until its cache is full.

*A. Capacity Based System Tthroughput Performance*

Fig. 4 illustrate the performance of our proposed scheme where we set and . Fig. 4 shows the effective capacity based system throughput when the num of users varies in the network. Obviously we see that as the number of user increases, the system throughout improves and the mobility-aware 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 transmition delay.

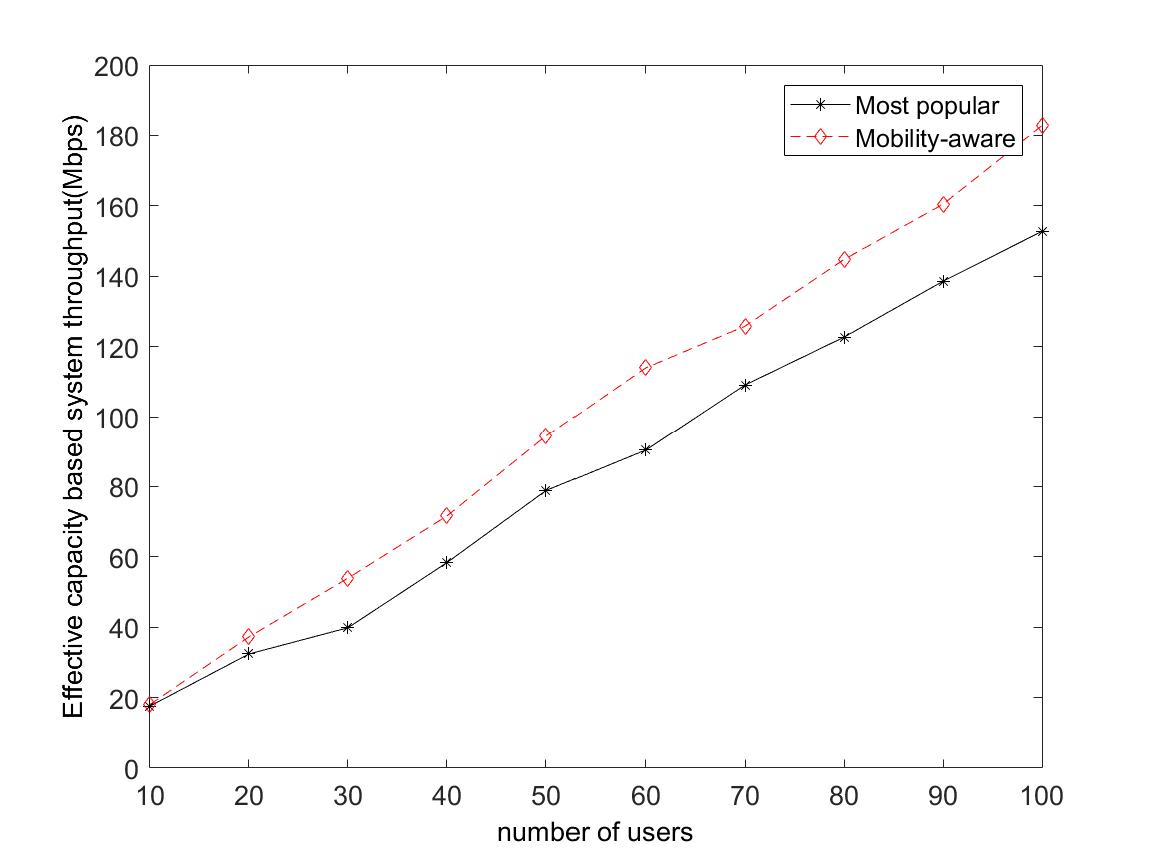


Fig. 4

Fig.5 investigate the relationship between cache size and system throghput. As shown in Fig. 5, with the total cache size increases, the effective capacity based system throughput gains 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.

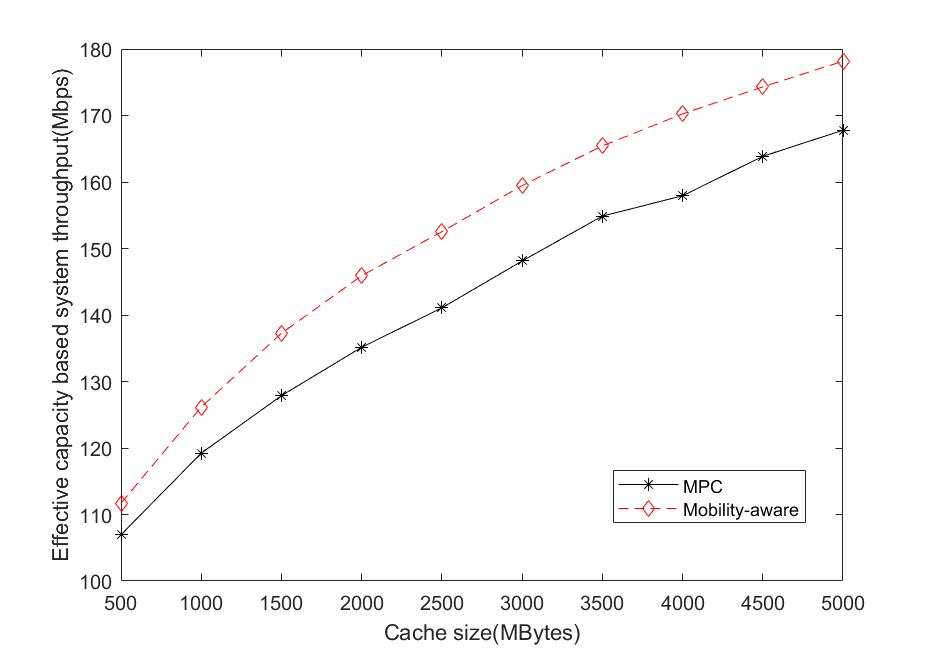


Fig. 5

*B. Cache Hit Ratio Performance*

Fig. 6 display the impact of file populatity distribution parameter on the cache hit ratio. In this simulation, user number is set to 70. As the figure shows, when is small, both two caching strategy have low cache hit ratio. Because in this case, all files are nearly equally likely to be requested but the storage capacity of BS is limited. But mobility-aware caching strategy still achieves superior performance than MPC. When 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.

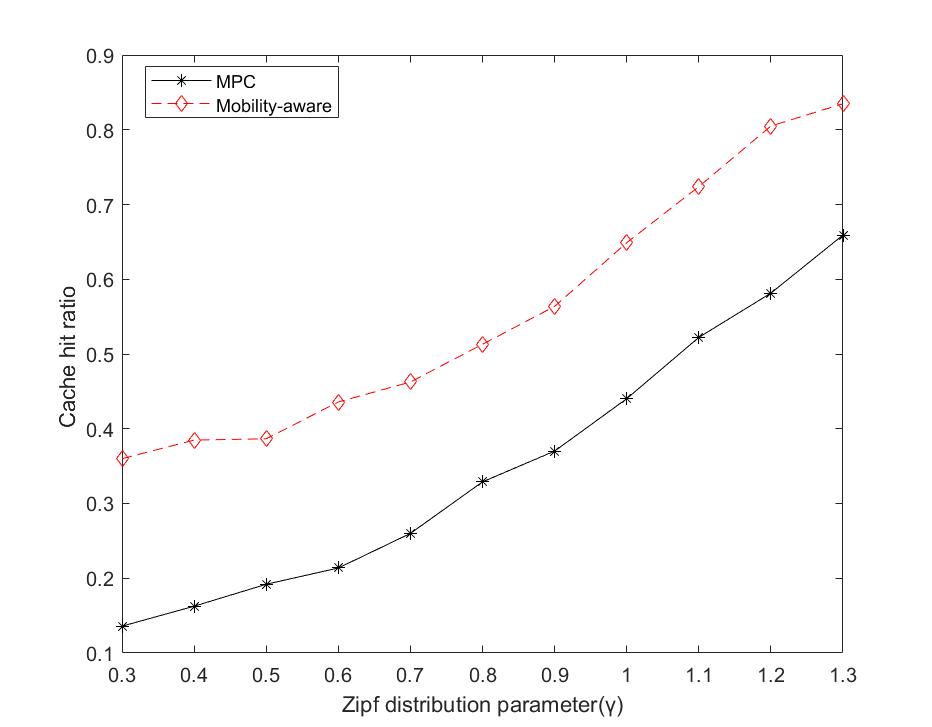


Fig.6

**VI.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. We use random jump model and stationary Markov model to describe the mobile pattern of user and amend the popularity at BSs with it. Then, we formulated the optimal content placements problem and solved it by genetic algorithm. 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. This indicates that our proposed caching strategy is a promising way to address the challenge of network densification.