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

**Abstract**

In this paper, we proposed a new mobility-a[[1]](#footnote-1)ware 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 as a 0-1 integer nonlinear programming 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.

**Keyword: Heterogeneous ultra-dense networks (HetUDNs); Mobility; Caching Strategy; 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,Cisco]. 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 [2,5G wireless backhaul networks: challenges and research advances].

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,Cache-enabled small cell networks: Modeling and tradeoffs], 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,Cache in the air: exploiting content caching and delivery techniques for 5G systems]. 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,FemtoCaching] 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,cluster hetnet coded] 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,Energy-efficient Cooperative Coded Caching for Heterogeneous Small Cell Networks], the authors propose an cooperative caching strategy where SBSs 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 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,Energy-Efficient Context-Aware Matching for Resource Allocation in Ultra-Dense Small Cells]. 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,Mobility-aware caching for content-centric wireless networks: modeling and methodology]. Reference [10,Mobility-Aware Coded Probabilistic Caching Scheme for MEC-Enabled Small Cell Networks] 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,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 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 macrio base station (MBS) [12,Code, cache and deliver on the move: a novel caching paradigm in hyper-dense small-cell networks].

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 requests that the SBS miss are redirected to the MBS [12,Code, cache and deliver on the move: a novel caching paradigm in hyper-dense small-cell networks]. 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 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:

1) We propose a mobility-aware hierarchical proactive caching strategy in HetUDNs.

2) 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 date rate.

3) 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 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 proacvite caching strategy and formulate 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 two-tied HetUDN comprising a single MBS and K densely deployed SBSs. 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 nearest 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 .

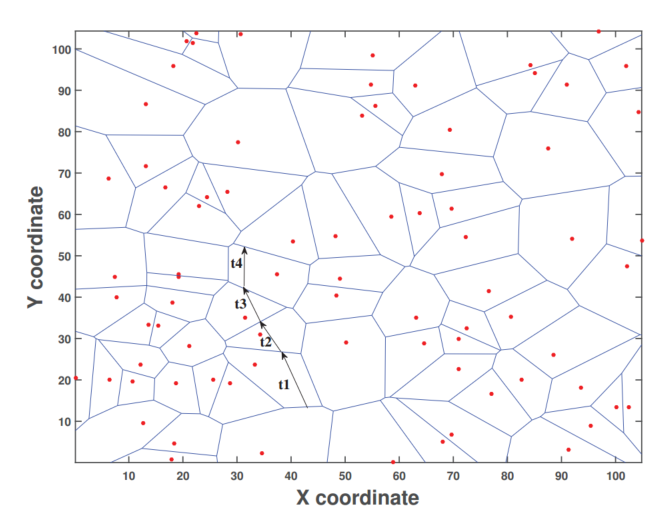


Fig.1 Illustration of a HetUDN with cache

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 with 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 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 , includes all the MBS and SBS. The cache storage capacity constraints can be expressed as:

1. *File Popularity Model*

We define the popularity 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 may not reflect the file request probability in the next time period.

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 [11,Mobility-Aware Coded Probabilistic Caching Scheme for MEC-Enabled Small Cell Networks]. 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.

1. *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,Effective capacity: a wireless link model for support of quality of service], 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.

According to the queuing theory, the queue delay of the packet satisfies the following relation [14, Effective capacity-based quality of service measures for wireless networks]*.*

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. [15,Octopus: A Cooperative Hierarchical Caching Strategy for Cloud Radio Access Networks ]

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]:

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 . 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. GA-based Mobility-aware Proactive Caching Strategy**

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

1. *Mobility-aware Proactive Caching Strategy*

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 popularity of content in BS may not reflect the actual request probability of content in BS in the next time period. 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 or other BSs. 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.

Consider a situation, 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 SBS is shared equally among the associated users of that SBS. Thus, the bandwidth allocated to each user connecting to SBS *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 constant 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 total effective capacity.

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 is a adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection, and it is inherently suitable for solving optimization problems with binary variables [16,Genetic algorithms: A survey].

Firstly, 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 calcutated through equation (\*). individuals with top objective values are chosen as elites and selected into next generation directly. The rest of the next generation population are generated through crossover and mutation operations. The crossover function select 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 number of individuals generated through crossover andmutation operations are denoted as and , respectively, where and the crossover fraction is . 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 {*X(K+1)F*}. For each row in each each individual, of the elemnets are set to be one randomly, and all the left elements are set to zero, where

For the storage capacity constraint in each BS.

1. Crossover Function: The crossover function use a two-point crossover function to generate two children, which is described in Algorithm 1. The steps 5-10 are heuristic process to meet constraint (##)
2. Mutation Funtion: 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 Np \* Ng , where Np and Ng are the population size and the number of generations evaluated, respectively.

Algorithm1: Crossover function

keep step 1~5 and 9~15

**V. 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 a 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 is, where is the skewness reflecting the concentration of the popularity distribution among users. All other simulation parameters are listed in Tabel 1.

|  |  |
| --- | --- |
| System Parameter | |
| Number of MBS | 1 |
| Number of SBSs *K* | 100 |
| Fronthaul rate | 600 Mbps |
| backhaul rate | 1000 Mbps |
| Bandwidth of SBS *G* |  |
| Number of Files *N* | 200 |
| Size of Files *B* | 50 MB |
| GA Parameter | |
| Population size Np | 50 |
| Terminal generation number Ng | 100 |
| Number of elites Ne | 10 |
| Crossover fraction fc | 0.8 |

Tabel 1. Simulation 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[5,FemtoCaching].

*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 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 transmition delay.

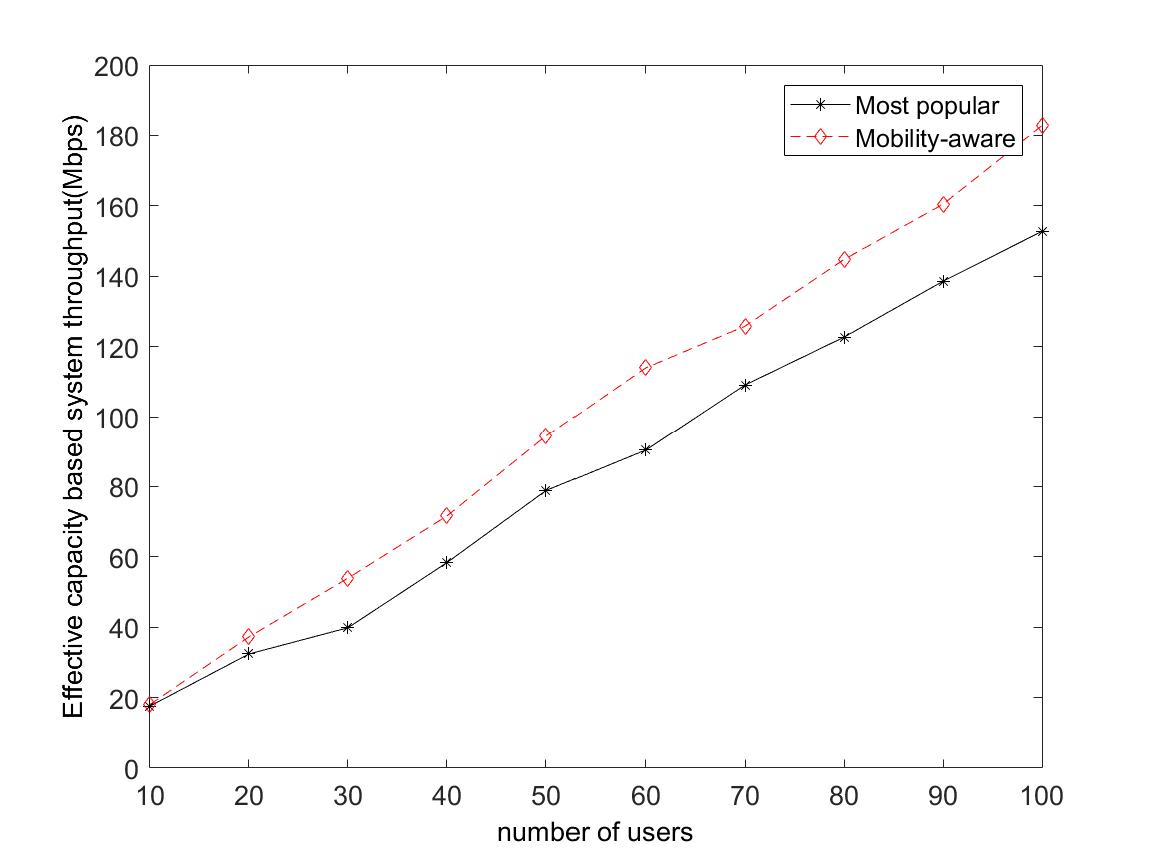


Fig. 2

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.

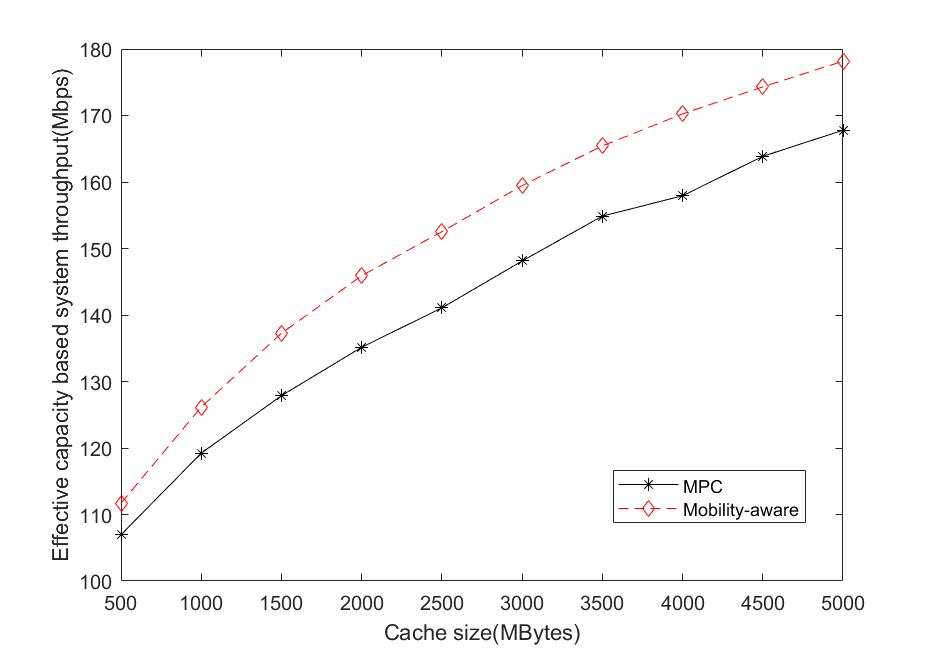


Fig. 3

*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 proactive 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.

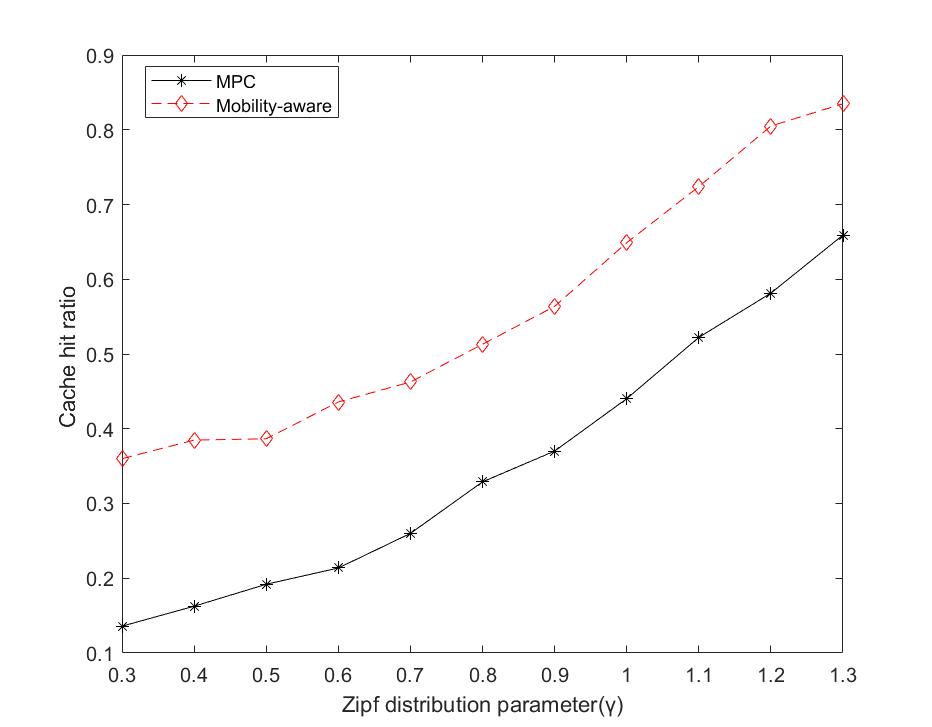


Fig.4

**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 as a 0-1 integer nonlinear programming 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.

1. [↑](#footnote-ref-1)