IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2020, accepted July 30, 2020, date of publication August 4, 2020, date of current version August 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014275

Time-Varying Mobile Edge Computing for Capacity Maximization

YUNYUN CAI AND PEIYAN YUAN, (Member, IEEE)
School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China

Corresponding author: Peiyan Yuan (peiyan@htu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1804164, and in part by the Henan Science and Technology Research Project under Grant 202102210323.

ABSTRACT Capacity is a fundamental metric for mobile edge computing scenarios, where the system state plays an important role. Previous studies have mostly been based on the premise that the system state is stable. In reality, the network is dynamic and the system state changes with time. In this paper, we study the capacity of a mobile edge system in which users continuously join or leave the coverage of base station. We first change the problem of maximum network capacity into a minimum transmission distance problem. We observe that both the probability of the files being requested and the distance of the files transmission are related to the degree of files, i.e., the number of users who are interested in a file and request it with a certain probability. Then, we evaluate the degree of files in a time-varying situation, and calculate the probability of the files being requested and the transmission distance according to the degree of files. Finally, we calculate the capacity of the network under time-varying conditions. In the experimental section, we analyze the degree of files, the optimal copies number, and the change in network capacity over time. In addition, we compare the capacity in our system with classic studies. The experimental results verify the superiority of the proposed method.

INDEX TERMS Mobile edge computing, capacity, time-varying, degree of files, transmission distance.

I. INTRODUCTION

According to Gartner's report, the number of devices connected to the Internet will reach 20.8 billion by 2020. The pervasive connections will urgently need a more competitive, scalable, secure and intelligent access network. Mobile edge computing (MEC) [1]–[3] provided a flexible platform by integrating storage, computing and communication into base stations (BSs) or small base stations (SBSs), which are closer to end users, to reduce access delay and achieve a better quality of experience [4]–[7]. There have been many studies on MEC so far [8]–[12].

Capacity indicates the ability of a channel to transmit signals that can reflect the maximum transmission rate supported by the channel. Gupta and Kumar [13] proved that each node can transmit at most $\theta\left(\frac{W}{\sqrt{n}}\right)$ bits per second, where n is the number of nodes, W is the channel throughput and θ is used to characterize the growth rate of the function. Evaluating capacity in MEC systems becomes more complex. First, the popularity of files vary at different moments and for different users. Second, the devices of the network contain BSs, SBSs, and users, and their functions/roles are different. Third, users are generally mobile rather than static. Some related work study capacity from different perspectives [14]–[21]. However, most of the papers on capacity cache files in nodes according to the popularity of the files (popularity follows Zipf distribution). While in practice, the popularity of files changes with users requests. For example, the authors of [22]–[24] allocated resource and cache files by effectively predicting the distribution of users' requests and its mobility model. In addition, [14]–[21] assumed that the scenario is stationary, failing to reflect the network dynamics. In reality, the system status can evolve over time. The network state can be stable or time-varying (unstable). In a stable state, the values of variables in the network are constant. In the time-varying (unstable) state, the variables in the network change with time. For example, the number of users, the location of users, the number of files and the popularity all change with time. When users join or leave the coverage of the BS, the number of users in the BS will change, which may make the probability of files being requested and the optimal copies number change over

The associate editor coordinating the review of this manuscript and approving it for publication was Alessio Vecchio.

time. Some papers analyzed the system performance with the time varying network state. For example, new nodes appear in the graph as time evolved, to reflect the fact that the web is changing with time [25]. The authors of [26] observed that most graphics density changes over time with the number of edges growing superlinearly in the number of nodes and the average distance between nodes often shrinking over time. In [27], [28], they found that in many applications, the popularity profile is unknown and changes over time; hence, they analyzed the cache with nonstationary and statistically dependent popularity profiles (hypothetically unknown and therefore estimated) from a learning theory perspective. Liu *et al.* [29], [30] proposed a novel evolving model in which the hybrid interactions among entities, based on whether they belong to the same type, are classified into intertype and intratype interactions that are characterized by two joint graphs evolving over time. The proposed model was verified through simulations, and the results show that the model can well capture realistic networks. The authors of [31] proved that the evolution of the network over time can increase the files delivery rate from the perspective of the degree of files, i.e., the number of users who are interested in the file and may request it with a certain probability. These studies are different from traditional static networks.

Recall that MEC provides services for local users, and the files in the edge cache are more targeted to users within their coverage. However, under time-varying conditions, the users within the coverage of the BS are constantly changing, and the users' preference for the files will also change, then the popularity of the files and the network capacity are change over time. Considering this fact, we study the capacity of MEC over time. Our contributions are as follows.

We mainly analyze the capacity of the edge network under time-varying conditions, which is different from other static network environments.

We use a bipartite graph to represent the evolutionary relationship between users and files, in which users and files are treated as two disjoint subsets, and each user and file is a vertex in bipartite graph.

We observe that the degree of the files plays a big role in system capacity, and calculate the degree of the files by the users' arriving and leaving. We then use the degree of the files to calculate the probability of files being requested and the transmission distance of the files.

We calculate the time-varying optimal copies number based on the probability of the files being requested, so that the users can request the files at a relatively close distance which can minimize the average transmission distance and maximize the capacity.

The rest of this paper is organized as follows. We survey related work in Sec. II. In Sec. III, we introduce the time-varying network model. In Sec. IV, we calculate the degree of files and the probability of the files being requested. In Sec. V, the minimum transmission distance is calculated by the degree of files and the probability that the files being requested, and analyze the copies number of files. In Sec. VI, we calculate capacity of the edge network and discuss the change in the degree of files. In Sec. VII, we analyze the degree of files, the copies number of files, and the change in network capacity over time and compare the capacity in our system with classic studies. We provide a brief summary of this paper in Sec. VIII.

II. RELATED WORK

There have been considerable researches on capacity. Most of the current studies are based on the assumption that the network state is stable [21], [32], [33]. In [14], the authors introduced a general class of mobile networks that incorporate both restricted mobility and inhomogeneous node densities and described a methodology to compute the asymptotic throughput achievable in these networks by the store-carry-forward communication paradigm. The authors of [15] studied the throughput capacity of an information-centric network when the data cached in each node has a limited lifetime and proved that increasing the files lifetime according to the network growth can enhance throughput capacities. In [16], the authors analyzed the effect of cooperation on network capacity in a hybrid network composed of cellular and device-to-device (D2D) communications, and their simulation results showed that the cooperation between cellular and D2D links can contribute to the sum capacity of the hybrid network. In [17], Yang *et al.* analyzed the transmission capacity of D2D communication under heterogeneous networks with cellular users (CeUEs) assisted and stated that the D2D network capacity can be enhanced by allocating a part of CeUE transmission power to assist D2D communication. Different from the previous research, the authors in [18] extended the exact capacity study for mobile ad hoc networks (MANETs). In [21], they proposed a simple model that captures two key characteristics observed in real large-scale networks, i.e., how people select friends and the number of friends, and examined their impact on capacity.

In [19], the authors proposed that storing part of the files in cache can increase network capacity. However, simply adding files to cache limits the capacity increase, and the amount of files cached is not allocated according to popularity, which results in cached files are not necessarily popular for users, leading to an increase in transmission distance and a decrease in capacity. The research on capacity in [20] significantly improved upon the research in [19]. The author distributed the files according to popularity, and each file was not only a copy in the network, i.e., more popular files had more copies. However, the calculation of transmission distance in this paper was based on [19]. The distribution of nodes in the network is uniform, which does not reflect reality. Additionally, as time changes, the files cached in the network may not be popular for the users. In this article we assume that the state of the users, the popularity of the files, and the maximum value of the network capacity all change over time.

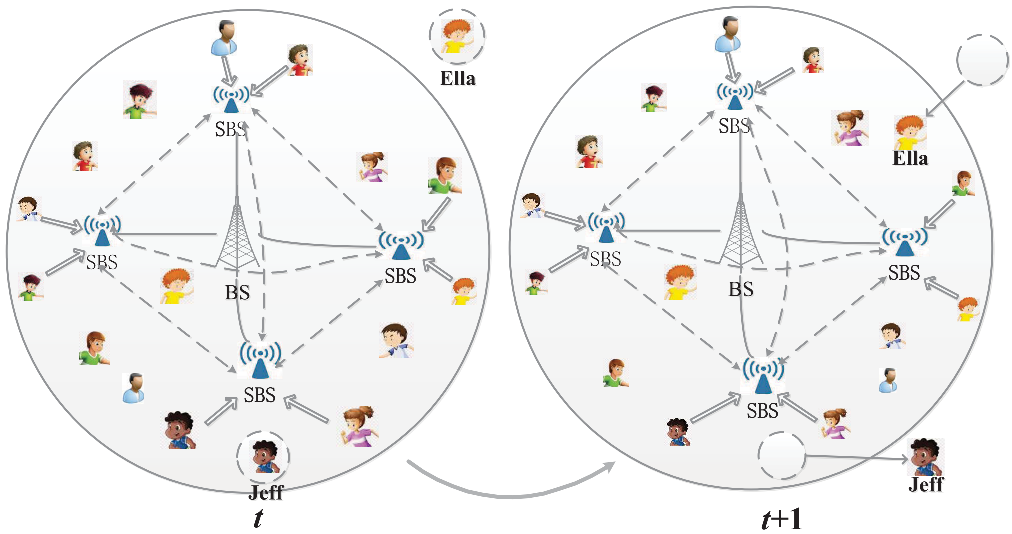


FIGURE 1. System architecture under time-varying conditions.

III. PRELIMINARIES

MEC can alleviate the bandwidth pressure and latency caused by the rapid development of the Internet and the Internet of Things. Hence, users can enjoy a higher quality of experience. However, under time-varying conditions, the users continuously arrive and leave the coverage of the BS, and the files initially cached may not necessarily be popular for the users. Therefore, the capacity of each moment can be calculated according to the latest statistics of the number of users and popularity. As such, we propose an edge computing model under time-varying conditions.

The system architecture, as shown in Fig. 1, includes one BS, n SBSs and $U(0)$ users, where the $U(0)$ denotes the initial number of users. Compared with a BS, the disadvantage of an SBS are smaller processing capacity, limited cache capacity (how many files an SBS can cache) and coverage and the advantages of smaller energy consumption and lower deployment cost. Within the coverage of a BS, there can be multiple SBSs, that is, there is a relatively close transmission distance between users and SBSs. In this paper, n SBSs are covered by the BS and SBSs can collaborate with each other to provide services to users. Under time-varying conditions, users may arrive or leave the BS coverage. We assume that all SBSs follow the Poisson Point Process (PPP) with an intensity of λ. PPP is one of the commonly used methods to place nodes. We assume that the BS can access the library of m files $F = \{F_1, \cdots, F_m\}$. Because different files have different sizes, we record the size of the jth file as b_j. Then we can assume that the average size of each file is b, where $b = \left(\sum_{j=1}^{m} b_j\right)/m$. Each SBS can cache an average of s files. Moreover, we assume that $C = \{C_1, C_2, \cdots, C_m\}$ is a placement vector and x_j^i indicates whether the jth file is placed in the ith SBS. $C_j(t)$ indicates the number of jth file placed in the entire edge network at the moment t, i.e., $C_j(t) = \sum_{i=1}^{n} x_j^i$.

Fig. 1 shows the arriving and leaving of users. At time t, Ella is outside BS coverage and Jeff is within BS coverage. Ella is within BS coverage but Jeff is outside BS coverage at time $t+1$. The number of users at time t is $U(t)$.

The authors of [31] indicated that users with strong social relations tend to request the similar files and the evolution between users and files can be represented by a bipartite graph. As shown in Fig. 2, where the set of users and files are two disjoint parts. Each user or file is equivalent to a vertex in the graph. Based on the bipartite graph, when user i is interested in file j and requests j with a certain probability, an edge is established between the user and the file, and the degree of the jth file also increases (the degree of file is the number of users who are interested in the file and will request the file with a certain probability). Assume that under the initial conditions, the number of edges per user is at least c_u, the number of edges per file is at least c_f, and there are at least $c_u c_f$ edges between users and files. As such, in the initial state, each user's degree is at least c_u, and each file is at least c_f. When a user q enters the coverage of the BS, the newly arrived user q selects a user q' who has the highest similarity with the new user as its prototype. For the calculation of similarity, the server stores the users' access records, which is similar to the users' browsing records in the cookie to facilitate users' tracking. When a new user enters the coverage of the BS, the types of file the new user are interested in are compared with the types of file existing users are interested in respectively. The similarity between the new user and the existing user is the ratio of the number of file types in the intersection and those in their union. For example, if the interested file types of one user are type A, type B, and type C, and the file types that another user are type B, type C, and type D, the similarity between the two is $\frac{n_{inter}}{n_{union}} = \frac{2}{4}$, where n_{inter} is the number of types in the intersection and n_{union} is the number of types in the union. New user copy the $S_{i,\kappa} d_i(t)$ ($d_i(t)$ is the degree of user i at

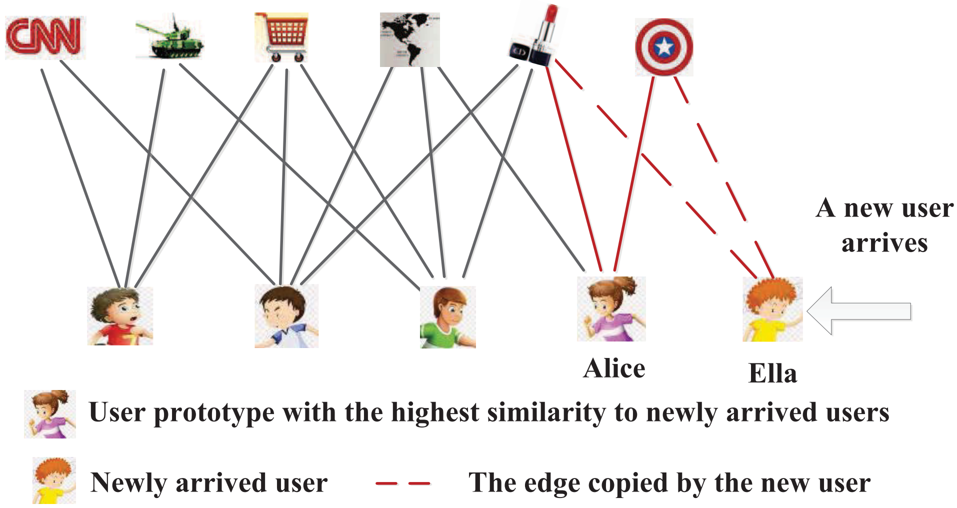


FIGURE 2. User's evolution graph.

time t) edges of the selected prototype, that is, the number of edges copied from the prototype is directly proportional to the similarity between the two users. In fact, this process is to make a preliminary judgment on the new users. If the similarity between the new user and all existing users is 0, the new user will be given c_u edge (this process is equivalent to the process of new user initialization). In Fig. 2, when Ella arrives, she chooses Alice as a prototype. They may have very similar requests for online information. For example, they may like movies, such as *Captain America*, *Hulk*, and *Spider-Man*. They may also be interested in cosmetics and beautiful clothes.

Now we consider the transmission of files. If one user wants to request a file, the user first sends the request to an SBS (the user will request the closest SBS and be within the coverage of the SBS). If the SBS has the file requested by the user, the SBS immediately provides it to the user; otherwise, the file request is forwarded to other SBSs. Generally, when an SBS provides file to one user, it may be provided to the user within its coverage or may be provided to the user within the coverage of other SBSs. If the file is delivered to user within the coverage of other SBSs, the file needs to be delivered to the SBS of the user originally request rather than delivered directly to the user.

Due to interference and noise in the network, the transmission of files in the network needs to meet certain conditions. Suppose all nodes use the same transmit power P, $V(t)$ denotes a set of SBSs that are transmit simultaneously at time t. When a node $i \in V(t)$ sends file to a node $j \in V(t)$, the transmission rate can reach W bits per second if the following conditions are met:

$$\frac{\frac{P}{d_{i,j}^{\varepsilon}}}{N_0 + \sum_{\varphi \in V(t), \varphi \neq i} \frac{P}{d_{i,j}^{\varepsilon}}} \geq \gamma \tag{1}$$

where γ is the minimum SINR that is successfully received, $d_{i,j}^{\varepsilon}$ is the distance between i and j, N_0 is white noise, and ε is a parameter greater than 2, which describes how the signal strength decays with distance.

In this paper, our main task is to calculate the capacity of MEC under time-varying conditions, that is, to maximize the network capacity when users continuously arrive or leave the BS coverage under time-varying conditions. We use the definition of capacity in [20], i.e., the number of bits each node utilizes per second to satisfy users' requests, which includes the number of files it receives from others and the files in the local cache that have been used to serve requests. It is indicated in [13], [19] and [20] that the relationship between capacity and transmission distance is $C = \frac{W}{L\sqrt{n}}$. Let $L(t)$ denote the average transmission distance required for the SBSs to deliver files to the users at time t, we can express the capacity as:

$$C(L(t)) = \max\left(\frac{W}{L(t)\sqrt{n}}\right) \tag{2}$$

From the expression of capacity, we can see that if we want to optimize the network capacity at time t, we need to minimize the transmission distance at time t. Since different files have different probability of being requested and there are m unique files in the system, we can transfer optimal goal from maximum capacity problem into minimum transmission distance as follow:

$$\min\left(L(t) = \sum_{j=1}^{m} L_j\left(C_j(t)\right) p_j(t)\right) \tag{3}$$

$$\text{s.t.} \begin{cases} C_j(t) \leq n \\ \sum_{j=1}^{m} C_j(t) \leq ns \end{cases} \tag{4}$$

where $L_j\left(C_j(t)\right)$ is the average transmission distance when user request for the jth file, and $p_j(t)$ is the probability that the jth file is requested at time t. The two constraints represent the copy number of each file in SBSs cannot exceed the number of SBSs and the copies number of all files in SBSs cannot exceed the volumes of the system, respectively. Note that the $L(t)$ includes two important factors, one is the probability of files being requested over time, and the other is the copies number of the files. Next, we will discuss the optimal copies number at the current time after calculating the probability of files being requested, so as to minimize the average transmission distance and maximize the capacity.

IV. THE PROBABILITY OF FILES BEING REQUESTED

We know that under time-varying conditions, there will be users constantly arriving or leaving the coverage of the BS. Different users will request different files, and the users' preference for files are different in different time periods. Then, the probability that one file is requested in the network is different in each time period. More popular files will be cached more often. Therefore, it is important to be able to accurately predict the probability of the files being requested. On the other hand, the degree of files is similar to the probability of the files being requested [31] and reflects the popularity of the files. Thus, the probability that the files being requested in this paper is calculated by the degree of files, which is different from the traditional Zipf distribution.

Lemma 1: In our time-varying network model, the degree of the jth file at time t can be expressed as:

$$d_j(t) = d_j(0) \prod_{l=0}^{t-1} \left(1 + \frac{e_{in}(t) - e_{out}(t)}{(e_{in}(t) - e_{out}(t))(l-0) + c_u c_f}\right) \tag{5}$$

where $e_{in}(t) = p_{k_{in}}(t) \sum_{k=1}^{k_{in}} S_{i,k} d_i(t)$, $e_{out}(t) = \sum_{i=1}^{k_{out}} p_{k_{out}}(t) d_i(t)$.

Proof: Suppose the number of users arriving in a unit of time may be $0, 1, 2, 3 \cdots$. The number of requests across users in any interval follows an independent Homogeneous Poisson distribution [28] and then we can record the arrival rate of users as λ_{in} following the Homogeneous Poisson distribution. Thus, the probability that the k_{in} users reach the BS coverage within the time interval t can be expressed as $p_{k_{in}}(t) = \frac{\lambda_{in}^{k_{in}}}{k_{in}!} e^{-\lambda_{in}}$, where $p_{k_{in}}(t) \geq 0$, $k_{in} = 0, 1, 2, \cdots$. Similarly, we can assume that the users' leave in a unit time follows the Homogeneous Poisson distribution with intensity λ_{out}, i.e., the probability that there are k_{out} users leaving the BS range within the time interval t can be expressed as $p_{k_{out}}(t) = \frac{\lambda_{out}^{k_{out}}}{k_{out}!} e^{-\lambda_{out}}$, where $p_{k_{out}}(t) \geq 0$, $k_{out} = 0, 1, 2, \cdots$.

Note that when a new user arrives in the coverage of the BS, the newly arrived user will select one user with the highest similarity as the prototype. The newly arrived user then copies the $S_{i,\kappa} d_i(t)$ edges of the selected prototype. Then, when a new user arrives, the new user may request jth file with the probability $\frac{d_j(t-1)}{e(t-1)}$, where $e(t-1)$ refers to the total number of all edges between files and the users, i.e., the sum of degrees of all files, and $d_j(t-1)$ is the degree of jth file at time t-1. That is, the larger the degree of the file, the greater the probability of the file will be requested by users.

When one user leaves the coverage of the BS, the edges associated with the user will break, i.e., the number of edges that leave are the degree of the users. Then, the probability that all related edges of the leaving user are disconnected from the jth file is $\frac{d_j(t-1)}{e(t-1)}$. Therefore, we can calculate the degree of files based on the change in the number of users arriving or leaving in a unit of time.

Therefore, the degree of the jth file at the time t interval is:

$$\begin{aligned} d_j(t) &= d_j(t-1) \\ &\quad + \left(p_{k_{in}}(t) \sum_{\kappa=1}^{k_{in}} S_{i,k} d_i(t) - \sum_{i=1}^{k_{out}} p_{k_{out}}(t) d_i(t)\right) \\ &\quad \times \frac{d_j(t-1)}{e(t-1)} \end{aligned} \tag{6}$$

where $p_{k_{in}}(t) \sum_{\kappa=1}^{k_{in}} S_{i,k} d_i(t)$ and $\sum_{i=1}^{k_{out}} p_{k_{out}}(t) d_i(t)$ refer to the number of edges join the network and the number of edges leave the network per unit time. Then, we can get:

$$\begin{aligned} d_j(t) &= d_j(t-1)\left(1 + \frac{e_{in}(t) - e_{out}(t)}{e(t-1)}\right) \\ &= d_j(t-1)\left(1 + \frac{e_{in}(t) - e_{out}(t)}{(e_{in}(t) - e_{out}(t))(t-1-0) + c_u c_f}\right) \\ &= d_j(0) \prod_{l=0}^{t-1} \left(1 + \frac{e_{in}(t) - e_{out}(t)}{(e_{in}(t) - e_{out}(t))(l-0) + c_u c_f}\right) \end{aligned} \tag{7}$$

Therefore, Lemma 1 is proven.

Theorem 1: Under our network model, the probability that the jth file is requested at time t can be calculated as:

$$p_j(t) = \frac{d_j(t)}{\sum_{j=1}^{m} d_j(t)} \tag{8}$$

Proof: From Lemma 1, the degree of the jth file at time t can be expressed as:

$$d_j(t) = d_j(0) \prod_{l=0}^{t-1} \left(1 + \frac{e_{in}(t) - e_{out}(t)}{(e_{in}(t) - e_{out}(t))(l-0) + c_u c_f}\right) \tag{9}$$

We know the degree of files is the same as the probability of the files being requested and reflects the popularity of the files, then probability of the files being requested can be expressed as:

$$p_j(t) = \frac{d_j(0) \prod_{l=0}^{t-1} \left(1 + \frac{e_{in}(t) - e_{out}(t)}{(e_{in}(t) - e_{out}(t))(l-0) + c_u c_f}\right)}{\sum_{j=1}^{m} d_j(0) \prod_{l=0}^{t-1} \left(1 + \frac{e_{in}(t) - e_{out}(t)}{(e_{in}(t) - e_{out}(t))(l-0) + c_u c_f}\right)}$$

$$= \frac{d_j(t)}{\sum_{j=1}^{m} d_j(t)} \tag{10}$$

Consequently, Theorem 1 is proven.

V. TRANSMISSION DISTANCE

Calculating the minimum transmission distance is a critical step in calculating the maximum network capacity. In this section, we first represent the average transmission distance of the files delivered to the users, and then we calculate the copies number of files that can minimize the transmission distance.

A. REPRESENTATION OF TRANSMISSION DISTANCE

Users in the coverage of all SBSs may request the jth file directly or indirectly. If an SBS delivers one file to the user, the probability that the SBS will deliver the file to the user within its coverage is $1/n$, and the probability of delivering it to other SBSs is $\frac{n-1}{n}$. Since the transmission distance from the SBS to users within its coverage is relatively small, we ignore it here. Then, the transmission distance can be expressed as:

$$\begin{aligned} L(t) &= \frac{1}{n} \cdot 0 + \frac{n-1}{n} \overline{L(t)} \\ &= \frac{n-1}{n} \sum_{j=1}^{m} \overline{L_j(C_j(t))} p_j(t) \end{aligned} \tag{11}$$

where $\overline{L_j(C_i(t))}$ is the average transmission distance from the SBSs that delivers the jth file to other SBSs.

Theorem 2: Under our network model, the average transmission distance over which SBSs deliver files to users can be calculated as:

$$E\left(\overline{L(t)}\right) = \sum_{j=1}^{m} \frac{d_j(t)\, p_j(t)}{2\sqrt{\lambda} C_j(t)(n-1)} \tag{12}$$

Proof: When calculating the transmission distance over which the SBSs delivers the jth file to the users, we start with the degree of files. In Sec. IV, we have calculated the degree of files, and the degree of the each file determines the number of users for each file service. That is, when one SBS delivers the jth file to other SBSs, it may deliver to one of $d_j(t)$ SBSs. Since each file has $C_j(t)$ copies at time t, the jth file delivered by SBS may be one of $C_j(t)$ files. Then, every file in $C_j(t)$ may be delivered to $\frac{d_j(t)}{C_j(t)}$ users, where $d_j(t) > C_j(t)$. In addition, it is assumed that X_j^i represents the transmission distance between the SBS receiving the request for jth file and the ith neighboring SBS. We can express the transmission distance for delivering the jth file to the users as:

$$E\left(\overline{L_j(C_j(t))}\right) = \frac{\sum_{j=1}^{\frac{d_j(t)}{C_j(t)}} X_j^i}{n-1} \tag{13}$$

Considering that there are m unique files in the network and each has a different probability of being requested, we can calculate the average transmission distance when SBSs deliver files to users as:

$$E\left(\overline{L(t)}\right) = \sum_{j=1}^{m} \frac{\sum_{j=1}^{\frac{d_j(t)}{C_j(t)}} X_j^i}{n-1} p_j(t) \tag{14}$$

In the case where the SBSs in the network follow the PPP, the distance between the SBS and the nearest SBS is (the detailed proof process of (15) can be found in the appendix):

$$\begin{aligned} E\left(X_j^i\right) &= \int_0^{\infty} x f(x) dx \\ &= 2\pi\lambda \int_0^{\infty} x^2 \cdot e^{-\lambda\pi x^2} dx \\ &= \frac{1}{2\sqrt{\lambda}} \end{aligned} \tag{15}$$

We can rewrite the transmission distance as:

$$E\left(\overline{L(t)}\right) = \sum_{j=1}^{m} \frac{d_j(t)\, p_j(t)}{2\sqrt{\lambda} C_j(t)(n-1)} \tag{16}$$

Theorem 2 represents the transmission distance when users request files. Since the problem of maximizing capacity can be transformed into the problem of minimum transmission distance. Then we need to find out the conditions under which the transmission distance can be minimized.

B. OPTIMAL COPIES NUMBER AND MINIMUM TRANSMISSION DISTANCE

In this section, our main task is to calculate the minimum transmission distance when SBSs delivers files to users and the copies number of files that can minimize the transmission distance. Based on the average transmission distance we calculated in Sec. V-A, we can express the optimization problem (minimum transmission distance) as follows:

$$\min \left(\sum_{j=1}^{m} \frac{d_j(t)\, p_j(t)}{2\sqrt{\lambda} C_j(t)(n-1)} \right) \tag{17}$$

$$\text{s.t.} \begin{cases} C_j(t) \le n \\ \sum_{j=1}^{m} C_j(t) \le ns \end{cases} \tag{18}$$

In (18), all variables except $C_j(t)$ are known. If the files requested by the users have a larger copies number, the users will receive the files at a relatively close distance. The value of $C_j(t)$ directly affects the value of the transmission distance. However, the size of the cache space of the SBSs are limited, and it is impossible to store all files in edge SBSs. Hence the files need to be allocated reasonably in limited storage space in order to minimize the transmission distance and maximize the capacity. Next, the optimal copies number that can minimize the transmission distance will be discussed.

Theorem 3: The optimal copies number at time t can be expressed as:

$$C_j(t) = \begin{cases} 0, & p_j(t) \le 0 \\ \sqrt{\dfrac{d_j(t)\,p_j(t)}{2\beta\sqrt{\lambda}(n-1)}}; & 0 < p_j(t) < \dfrac{2\beta\sqrt{\lambda}(n-1)\,n^2}{d_j(t)} \\ n; & p_j(t) \ge \dfrac{2\beta\sqrt{\lambda}(n-1)\,n^2}{d_j(t)} \end{cases} \tag{19}$$

Proof: According to the standard form of the generalized Lagrange function, we have:

$$\sum_{j=1}^{m} \frac{d_j(t)\,p_j(t)}{2\sqrt{\lambda}C_j(t)(n-1)} + \sum_{j=1}^{m} \alpha_j\left(C_j(t)-n\right) + \beta\left(\sum_{j=1}^{m} C_j(t) - ns\right) \tag{20}$$

where α_j and β are the Lagrangian constants.

The KKT (Karush-Kuhn-Tucher) conditions are as follows:

$$\begin{cases} \sum_{j=1}^{m} \alpha_j\left(C_j(t)-n\right) = 0; & \alpha_j \ge 0 \\ \beta\left(\sum_{j=1}^{m} C_j(t) - ns\right) = 0; & \beta \ge 0 \end{cases} \tag{21}$$

Then, we can construct multivariate partial differential equations of $C_j(t)$:

$$\frac{\partial \bar{L}}{\partial C_i} = \frac{-d_j(t)\,p_j(t)}{2\sqrt{\lambda}C_j^2(t)(n-1)} + \alpha_i + \beta \tag{22}$$

Let the above formula equal 0, we obtain:

$$C_j(t) = \sqrt{\frac{d_j(t)\,p_j(t)}{2\left(\alpha_j+\beta\right)\sqrt{\lambda}(n-1)}} \tag{23}$$

Considering that the copies number of files in the network cannot be negative and then combine with the KKT conditions, we obtain:

$$\sqrt{\frac{d_j(t)\,p_j(t)}{2\left(\alpha_j+\beta\right)\sqrt{\lambda}(n-1)}} = 0 \Rightarrow p_j(t) = 0 \tag{24}$$

Since each file can have only one copy in one SBS and there are only n SBSs in the network, the copy number of each file in the network cannot be more than n.

Combining this cache constraint with the KKT conditions, we can obtain:

$$\sqrt{\frac{d_j(t)\,p_j(t)}{2\left(\alpha_j+\beta\right)\sqrt{\lambda}(n-1)}} = n \Rightarrow p_j(t) = \frac{2\beta\sqrt{\lambda}(n-1)\,n^2}{d_j(t)} \tag{25}$$

where $\alpha_j = \frac{d_j(t)p_j(t)}{2\sqrt{\lambda}(n-1)n^2} - \beta$.

In summary, the optimal copies number in Theorem 3 can be proved. Theorem 3 represents the copies number. This number is calculated according to the popularity of files, that is, the popular files have more copies, and vice versa. If files are stored in SBSs according to this standard at a certain moment, the average transmission distance of the files requested by the users is smallest, and (12) can obtain the minimum value.

Because the transmission distance of the files includes the transmission distance of the SBSs to the users within its coverage and the transmission distance to users in other SBSs, combing with (11), we can express the transmission distance as follows:

$$\begin{aligned} L(t) &= \frac{1}{n}\cdot 0 + \frac{n-1}{n}\bar{L}(t) \\ &= \frac{n-1}{n}\sum_{j=1}^{m} \frac{d_j(t)\,p_j(t)}{2\sqrt{\lambda}C_j(t)(n-1)} \\ &= \frac{1}{2\sqrt{\lambda}n}\sum_{j=1}^{m} \frac{d_j(t)\,p_j(t)}{C_j(t)} \end{aligned} \tag{26}$$

where

$$C_j(t) = \begin{cases} 0; & p_j(t) \le 0 \\ \sqrt{\dfrac{d_j(t)\,p_j(t)}{2\beta\sqrt{\lambda}(n-1)}}; & 0 < p_j(t) < \dfrac{2\beta\sqrt{\lambda}(n-1)\,n^2}{d_j(t)} \\ n; & p_j(t) \ge \dfrac{2\beta\sqrt{\lambda}(n-1)\,n^2}{d_j(t)} \end{cases} \tag{27}$$

VI. NETWORK CAPACITY

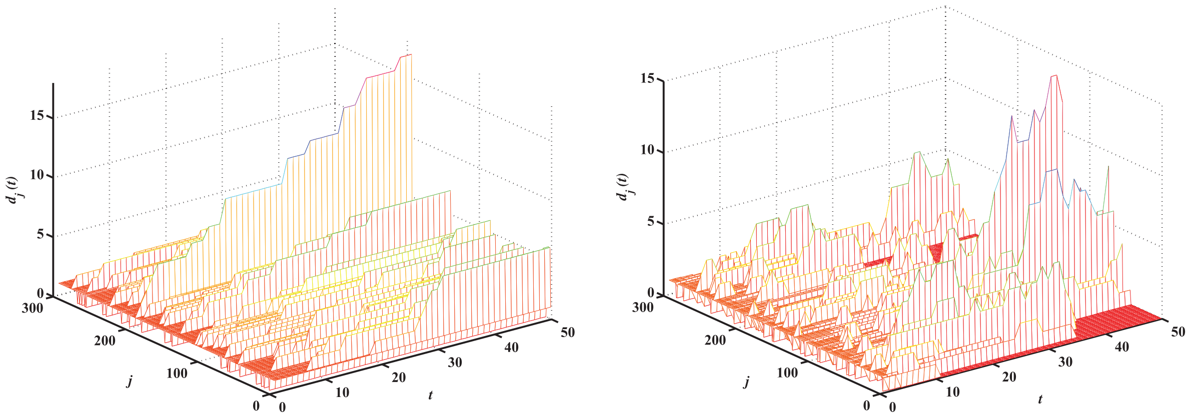
A. NETWORK CAPACITY

According to the average transmission distance obtained in Sec. V and combine with the calculation formula of the capacity of $\frac{W}{L(t)\sqrt{n}}$, we can calculate the network capacity as follows:

$$C(L(t)) = \frac{W}{L(t)\sqrt{n}} = \frac{W}{\frac{1}{2\sqrt{\lambda}n}\sum_{j=1}^{m}\frac{d_j(t)p_j(t)}{C_j(t)}\sqrt{n}} = \frac{2W\sqrt{\lambda}\sqrt{n}}{\sum_{j=1}^{m}\frac{d_j(t)p_j(t)}{C_j(t)}} \tag{28}$$

where

$$C_j(t) = \begin{cases} 0; & p_j(t) \le 0 \\ \sqrt{\dfrac{d_j(t)\,p_j(t)}{2\beta\sqrt{\lambda}(n-1)}}; & 0 < p_j(t) < \dfrac{2\beta\sqrt{\lambda}(n-1)\,n^2}{d_j(t)} \\ n; & p_j(t) \ge \dfrac{2\beta\sqrt{\lambda}(n-1)\,n^2}{d_j(t)} \end{cases} \tag{29}$$



(a) $\lambda_{uin} = 6, \quad \lambda_{uout} = 0$

(b) $\lambda_{uin} = 6, \quad \lambda_{uout} = 6$

FIGURE 3. The degree of files under time-varying conditions.

B. DISCUSSION

Combining with the above formula and reality, we can discuss in terms of users' arriving and leaveing as follows:

(a) When the number of users who arrive is not less than the number of users who leave ($K_{in} \geq K_{out}$), there are two variations in the degree of files. In the first case, only the users arrive the network, and no users leave ($K_{in} > 0$, $K_{out} = 0$). In this case, the degree of the jth file increases with a certain probability for each user's arrival. Since no user leave the network, the degree of all files is not reduced. Some files are not popular, and their degree may not change. Therefore, the degree of files will not decrease when only users arrive and no users leave. The second is the case where both the users arrive and the users leave ($K_{in} > 0$, $K_{out} > 0$). At this point, the degree of files may increase, unchange, or decrease. When the one user arrives at the network and requests certain file in the network with a certain probability, the degree of the requested file will increase with a certain probability. When a user leaves the network, the edge between the user and the file is broken, and the degree of file will decrease. There is also a case where some users' arriving and leaving are related to some files in the network. Overall, it is difficult to estimate whether the degree of files will increase, decrease or remain unchanged. Therefore, we cannot infer how the degree of the jth file changes.

(b) The second case is when the number of users arriving and leaving is equal to zero ($K_{in} = K_{out} = 0$). In this case, the degree of files, the popularity of the files, and the network capacity do not change. At this time, the network state is equivalent to being stable rather than time-varying.

(c) When the number of users leaving is larger than the number of users arriving ($K_{in} < K_{out}$), this situation may cause the number of users become to 0. We will not discuss this case in depth.

VII. NUMERICAL RESULTS

In the previous sections, we theoretically analyze the impact of the users' arriving and leaving on the degree, popularity, and capacity under time-varying conditions. In this section, we first simulate the change in the degree of the files when there are only users arriving without users leaving, and both users arriving and leaving. Then we analyze the change in copies number under time-varying conditions. Finally, the influence of some parameters on capacity is studied and the capacity of the proposed scheme is compared with the classic capacity in other papers. When calculating the capacity, we take the logarithm of the capacity in order to clearly show the relationship between the capacity. In addition, we perform four repetitions in the simulation phase and take their average as experimental results.

A. DEGREE OF FILES AND COPIES NUMBER

Fig. 3 (a) shows the change in the degree of the files over time, where the distribution intensity of SBSs is 50. The number of files is 300, and the number of users at the initial time is 50, which varies with time. We assume that there are only users arriving per unit time and no users leaving. We can see that the degree of each file is rising with different levels over time. Whenever a new user arrives the network, it will inevitably choose a user with high similarity as its prototype. The higher the similarity, the more edges will be copied, indicating that the files they request are more similar. Then the user will request a file with a certain probability, and the file with a higher degree will be requested by the users with a greater probability. In the simulation, the users use the Roulette Wheel Selection [32] to select files. Each file is assigned a probability of being selected according to the degree of the file. The greater the degree of the file, the greater the probability of the file being requested by the users, which is equivalent to the larger area occupied in the roulette. Therefore, the faster the growth of the files degree indicates that these files are liked by more users in the same period of time, the contrary indicates that the popularity of these files are not very high. Since there are no users leaving, the edges between the users and the files will not break.

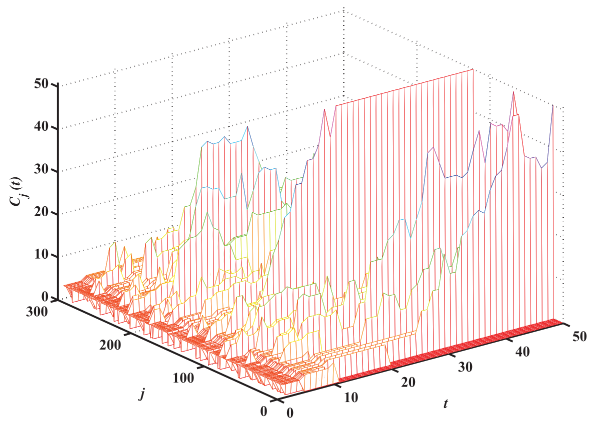


FIGURE 4. **The copies number of files under time-varying conditions.**

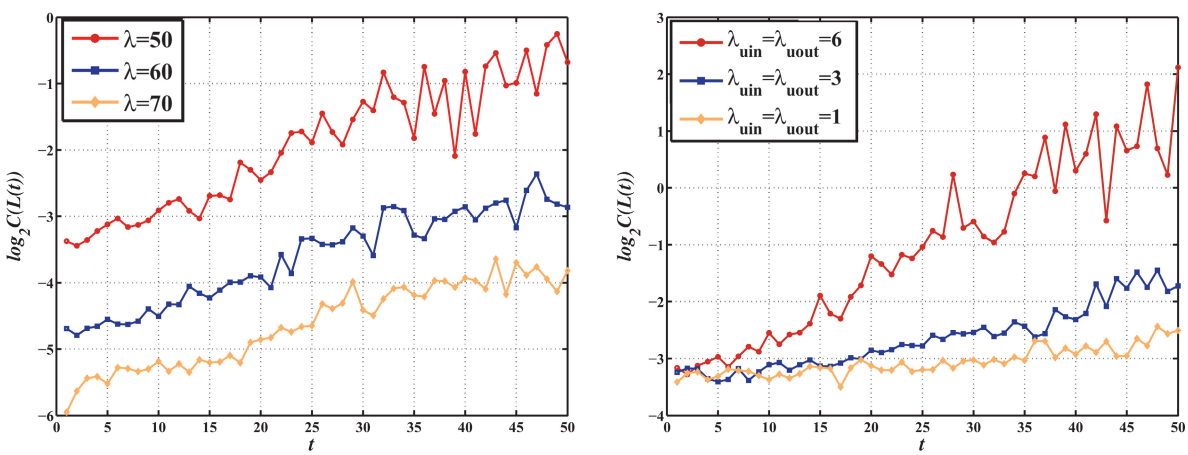
On the other hand, it is obviously impractical to assume that only users arriving without users leaving. So Fig. 3 (b) shows the change of the degree of the files as users arrive and leave. We set the number of arriving users is equal to the number of leaving users. Since the arriving users and leaving users are different users, the edge between the newly added users and the files and the edge between the leaving users and the files are also different. We can see that the degree of the files may rise or fall over time. The arriving of the users will increase the degree of the files. The users' leaving may reduce the degree of files. Of course, some files are always popular whether users arrive or leave. Because of the constant flow of users, files can be divided into several categories based on their popularity: a) files that have been popular for a long time; b) files that have been popular for a short time; c) files with moderate popularity; and d) files that have never been very popular. Therefore, it is obviously not feasible to calculate the capacity of system according to a fixed popularity.

Fig. 4 shows how the copies number of files change with time and popularity, where $C_j(t)$ denotes the optimal copies number of the jth file at a moment t. Since at the initial moment we assume that each file keeps the same degree, the popularity and copy number of each file will also be the same. As shown in Fig. 2, when a user enters the coverage of a BS, the user will be associated with the interested files. With the arriving and leaving of users, the difference in the request probability of different files becomes more obvious, and so does the difference in the degree of files. As time goes by, therefore, different files have different copies number.

B. ANALYSIS OF NETWORK CAPACITY

The two graphs in Fig. 5 show capacity under time varying conditions. Fig. 5 (a) analyzes the influence of the distribution intensity of the SBSs on the capacity, where $\lambda_{uin} = \lambda_{uout} = 6$. We take three different λ values. It can be seen that the larger the value of λ, the smaller the value of capacity. The increase of λ means the increase of SBSs, the interference between SBSs will also increase, and the probability of the files being successfully transferred will be reduced. An increase in the number of failed transmissions will result in an increase in transmission distance. Therefore, an increase in λ will cause a decrease in capacity. What's more, when one user enters the coverage of SBS, the user will request the interested file, which is equivalent to participating in the evaluation of files popularity. With arriving and leaving of different users, the popularity of files will keep changing. Note that the probability of files is to proportional their popularity, and users can retrieve files at a relative distance if the changing of popularity can be reflected, which will increase the capacity.

Fig. 5 (b) shows the impact of users arriving and leaving on capacity, and we set $\lambda = 50$. From the analysis of the degree and copies number of the files, we have learned that the popularity of the files will change as the users arrive or leave. We set up three sets of values regarding users arriving and leaving rates: $\lambda_{uin} = \lambda_{uout} = 6$, $\lambda_{uin} = \lambda_{uout} = 3$ and $\lambda_{uin} = \lambda_{uout} = 1$. Here we set the users' arriving rate



(a) λ vs. $C(L(t))$ (b) $\lambda_{uin}/\lambda_{uout}$ vs. $C(L(t))$

FIGURE 5. **Network capacity under time-varying conditions.**

and leaving rate to be the same value, because the number of users basically remain stable despite the arriving and leaving of users within a period of time. We find that the higher the arriving rate and leaving rate per unit time, the larger the capacity; otherwise, the smaller. If a user is interested in a file and establishes contact with it, it is equivalent to the user participating in a files' popularity evaluation. More users arriving or leaving means that more users are evaluating popularity, just like everyone will choose what they want. As a result, the difference in popularity between files will become more apparent with user participation. So SBSs will cache more files that users really need, which will increase the capacity.

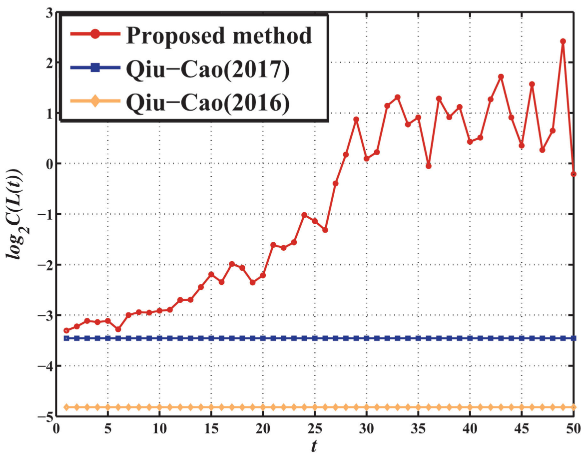


FIGURE 6. **Comparison of network capacity.**

In Fig. 6 we compare the proposed scheme with the classic schemes [19], [20], where $\lambda_{uin} = \lambda_{uout} = 6$ and $\lambda = 50$. It is clear that our capacity increases over time and the others do not. The capacity of our solution at the initial moment is not very good, because at the initial moment, the BS does not know the popularity of the files and cannot cache the files that are popular to the users. Although the value at the initial moment is not too satisfactory, it is more realistic. The other two capacity values are actually the network capacity values at a certain moment after the Zipf distribution is assumed.

From the above, our proposed scheme performs well, it greatly improves the network capacity compared with other schemes.

VIII. CONCLUSION

In this paper, we study the network capacity under time-varying conditions. To the best of our knowledge, this is the first time that files are cached in SBSs in order to maximize capacity through user mobility under time-varying conditions. First, we change the capacity maximization problem into a transmission distance minimization problem. We calculate the degree of files, and we find that using the degree of files, we can calculate the average transmission distance and the probability of files being requested by users. Then, we use the Lagrange multiplier to optimize the transmission distance and determine the optimal copies number that can minimize the transmission distance. Finally, we calculate the network capacity and implement simulation experiments. We first simulate the change in the degree of the files when there are only users arriving without users leaving, and both users arriving and leaving. Then we analyze the change in copies number under time-varying conditions. In addition, we compare our calculated capacity with the previous capacity. The results show that the capacity calculated by us shows superiority. Although our proposed scheme performs well, there are still some work that need further research in the future. First, we calculate the capacity based on the latest popularity and copies number at each moment. However, these files cannot be updated frequently with the latest updated data, because frequent updates will bring a lot of cost. And, if it is not updated for a long time, the cached files in SBSs will no longer be popular. So we need to find a balance in the cache update time. Second, we need to consider how network performance changes when congestion occurs.

APPENDIX
PROOF OF THE (15)

Proof: In the following, we use the knowledge about gamma functions to prove (15).

$$\begin{cases} \Gamma(\alpha) = \int_0^{+\infty} x^{\alpha-1} e^{-x} dx \\ \Gamma(\alpha+1) = \alpha\Gamma(\alpha) \\ \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi} \end{cases} \tag{30}$$

Now, we need to solve $\int_0^\infty x^2 \cdot e^{-\lambda\pi x^2} dx$. Let $t = x^2$. We have:

$$\begin{aligned} &\int_0^\infty x^2 \cdot e^{-\lambda\pi x^2} dx \\ &= \int_0^\infty t \cdot e^{-\lambda\pi t} \frac{1}{2\sqrt{t}} dt \\ &= \frac{1}{2\lambda\pi\sqrt{\lambda\pi}} \int_0^\infty (\lambda\pi t)^{\frac{1}{2}} \cdot e^{-\lambda\pi t} d\lambda\pi t \\ &= \frac{1}{2\lambda\pi\sqrt{\lambda\pi}} \Gamma\left(\frac{3}{2}\right) = \frac{1}{2\lambda\pi\sqrt{\lambda\pi}} \frac{1}{2} \Gamma\left(\frac{1}{2}\right) \\ &= \frac{1}{2\lambda\pi\sqrt{\lambda\pi}} \frac{\sqrt{\pi}}{2} \end{aligned} \tag{31}$$

Then,

$$E\left(X_j^i\right) = 2\pi\lambda \cdot \frac{1}{2\pi\lambda\sqrt{\lambda}} \cdot \frac{1}{2} = \frac{1}{2\sqrt{\lambda}} \tag{32}$$

Therefore, (15) is proven.

REFERENCES

[1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing-A key technology towards 5G," *ETSI White Paper*, vol. 11, pp. 1–16, 2015.
[2] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proc. 6th Int. Conf. Adv. Future Internet*, 2014, pp. 48–54.

[3] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[4] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.

[5] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[6] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 207–215.

[7] X. Lyu, H. Tian, L. Jiang, A. Vinel, S. Maharjan, S. Gjessing, and Y. Zhang, "Selective offloading in mobile edge computing for the green Internet of Things," *IEEE Netw.*, vol. 32, no. 1, pp. 54–60, Jan. 2018.

[8] P. Yuan, Y. Cai, X. Huang, S. Tang, and X. Zhao, "Collaboration improves the capacity of mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10610–10619, Dec. 2019.

[9] A. H. Sodhro, Z. Luo, A. K. Sangaiah, and S. W. Baik, "Mobile edge computing based QoS optimization in medical healthcare applications," *Int. J. Inf. Manage.*, vol. 45, pp. 308–318, Apr. 2019.

[10] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge computing based on Markov decision process," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1272–1288, Jun. 2019.

[11] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 10–18.

[12] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching," *IEEE Internet Things J.*, early access, Apr. 9, 2020, doi: 10.1109/JIOT.2020.2986803.

[13] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[14] M. Garetto, P. Giaccone, and E. Leonardi, "Capacity scaling in ad hoc networks with heterogeneous mobile nodes: The subcritical regime," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1888–1901, Dec. 2009.

[15] B. Azimdoost, C. Westphal, and H. R. Sadjadpour, "On the throughput capacity of information-centric networks," in *Proc. 25th Int. Teletraffic Congr. (ITC)*, Sep. 2013, pp. 1–9.

[16] Z. Liu, T. Peng, B. Peng, and W. Wang, "Sum-capacity of D2D and cellular hybrid networks over cooperation and non-cooperation," in *Proc. 7th Int. Conf. Commun. Netw. China*, Aug. 2012, pp. 707–711.

[17] Y. Yang, Z. Liu, Z. Fu, T. Peng, and W. Wang, "Transmission capacity of device-to-device communication under heterogeneous networs with cellular users assisted," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 635–641.

[18] Y. Chen, Y. Shen, J. Zhu, X. Jiang, and H. Tokuda, "On the throughput capacity study for aloha mobile ad hoc networks," *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1646–1659, Apr. 2016.

[19] L. Qiu and G. Cao, "Cache increases the capacity of wireless networks," in *Proc. IEEE INFOCOM 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[20] L. Qiu and G. Cao, "Popularity-aware caching increases the capacity of wireless networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 173–187, Jan. 2020.

[21] L. Fu, W. Huang, X. Gan, F. Yang, and X. Wang, "Capacity of wireless networks with social characteristics," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1505–1516, Feb. 2016.

[22] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized Quality-of-Experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[23] M. Chen, W. Saad, and C. Yin, "Liquid state machine learning for resource and cache management in LTE-U unmanned aerial vehicle (UAV) networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1504–1517, Mar. 2019.

[24] 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.

[25] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, "Stochastic models for the Web graph," in *Proc. 41st Annu. Symp. Found. Comput. Sci.*, Nov. 2000, pp. 57–65.

[26] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discovery from Data*, vol. 1, no. 1, p. 2, Mar. 2007.

[27] B. N. Bharath, K. G. Nagananda, D. Guenduez, and H. V. Poor, "Learning-based content caching with time-varying popularity profiles," in *Proc. GLOBECOM Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[28] B. N. Bharath, K. G. Nagananda, D. Gunduz, and H. V. Poor, "Caching with time-varying popularity profiles: A learning-theoretic perspective," *IEEE Trans. Commun.*, vol. 66, no. 9, pp. 3837–3847, Sep. 2018.

[29] J. Liu, Y. Yao, X. Fu, L. Fu, X.-Y. Liu, and X. Wang, "Evolving K-graph: Modeling hybrid interactions in networks," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jul. 2017, pp. 1–2.

[30] J. Liu, L. Fu, Y. Yao, X. Fu, X. Wang, and G. Chen, "Modeling, analysis and validation of evolving networks with hybrid interactions," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 126–142, Feb. 2019.

[31] Z. Qin, X. Gan, L. Fu, X. Di, J. Tian, and X. Wang, "Content delivery in cache-enabled wireless evolving social networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6749–6761, Oct. 2018.

[32] N. K. Meena, A. Swarnkar, J. Yang, N. Gupta, and K. R. Niazi, "Modified taguchi-based approach for optimal distributed generation mix in distribution networks," *IEEE Access*, vol. 7, pp. 135689–135702, Sep. 2019.

[33] L. Fan, Z. Dong, and P. Yuan, "The capacity of device-to-device communication underlaying cellular networks with relay links," *IEEE Access*, vol. 5, pp. 16840–16846, 2017.

YUNYUN CAI received the bachelor's degree in the Internet of Things engineering from the Xinlian College, Henan Normal University. She is currently pursuing the master's degree with the School of Computer and Information Engineering, Henan Normal University. Her research interest includes mobile edge computing.

PEIYAN YUAN (Member, IEEE) received the B.S. degree in computer science from Henan Normal University, Xinxiang, China, the M.S. degree in computer science from the Wuhan University of Technology, Wuhan, China, and the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications, Beijing, China. He worked as a Postdoctoral Researcher with the University of Texas at Dallas, USA. He is currently an Associate Professor of computer science with Henan Normal University. He has authored or coauthored more than 50 papers and one book in these fields. His research interests include future networks and distributed systems. He is a Senior Member of CCF and a member of ACM. He received the national scholarship for Ph.D. students from the Ministry of Education of China, in 2012, and the Best Paper Award of IEEE CSE, in 2014.