Peer-to-Peer Networking and Applications (2020) 13:1839–1852
https://doi.org/10.1007/s12083-020-00954-x

Content caching based on mobility prediction and joint user Prefetch in Mobile edge networks

Genghua Yu[1] · Jia Wu[1]

Received: 25 July 2019 / Accepted: 25 June 2020 / Published online: 8 July 2020

Abstract

With the development of 5G mobile networks, people's demand for network response speed and services has increased to meet the needs of a large amount of data traffic, reduce the backhaul load caused by frequently requesting the same data (or content). The file is pre-stored in the base station by the edge device, and the user can directly obtain the requested data in the local cache without remotely. However, changes in popularity are difficult to capture, and data is updated more frequently through the backhaul. In order to reduce the number of backhauls and provide caching services for users with specific needs, we can provide proactive caching with users without affecting user activity. We propose a content caching strategy based on mobility prediction and joint user prefetching (MPJUP). The policy predicts the prefetching device data by predicting the user's movement position with respect to time by the mobility of the user and then splits the partial cache space for prefetching data based on the user experience gain. Besides, we propose to reduce the backhaul load by reducing the number of content backhauls by cooperating prefetch data between the user and the edge cache device. Experimental analysis shows that our method further reduces the average delay and backhaul load, and the prefetch method is also suitable for more extensive networks.

Keywords Mobile edge cache · Popularity-based caching · Backhaul · Mobility predictions · Prefetching data

1 Introduction

With the development of mobile internet technology, a variety of mobile applications and multimedia services enrich people's lives [1], but also generate huge mobile network traffic, these applications and services rely heavily on high-rate and low-latency data transmission. According to the 2017 Cisco VNI Technical Report [3], global mobile data traffic will reach 587 EB by 2021, which is equivalent to 122 times in 2011. From 2016 to 2021, mobile video will grow by 8.7 times, accounting for 78% of total mobile traffic. The rapid growth of mobile network traffic, especially mobile video traffic, has brought tremendous pressure and challenges to the current mobile network.

In the 5G network era, users' demand for data has grown like never before. The proliferation of mobile network traffic has made bandwidth resources very tight. At the same time, the current end-to-end transmission mechanism causes a large amount of content to be repeatedly transmitted [2], resulting in a waste of network resources. In order to adapt to the rapid growth of network traffic, it is expected to be densely deployed by small cell base stations to meet a large number of data requests. As a result, the capacity of the base station is reduced, and the deployment of the wireless access point becomes denser, which will bring new opportunities for high-speed data transmission and content caching. However, due to the dense distribution of base stations, the cost of deploying backhaul for each small base station is high, making such an approach impractical. That is to say, the concentration of the mobile network architecture and the limited transmission capacity brought by the wireless backhaul link make this method unable to keep up with the explosive growth of traffic [4].

This article is part of the Topical Collection: *Special Issue on Emerging Trends on Data Analytics at the Network Edge*
Guest Editors: Deyu Zhang, Geyong Min, and Mianxiong Dong

✉ Jia Wu
jiawu5110@163.com

[1] School of computer science and engineering, Central South University Chang Sha, Hu Nan 410075, China

In order to cope with the growth of mobile network traffic and meet the millisecond delay of 5G networks, Mobile Edge Computing (MEC) is introduced into cloud computing. MEC servers are deployed on base stations (BSs) and nodes to provide computing and caching services to nearby users. The conventional scheme [5] is to store the data content requested by the user in a base station (BS) or an edge device so that the user can obtain the requested content directly in the base station closest to himself without requesting data from the remote data center when requesting the content. This can reduce data transmission delay and alleviate network bandwidth pressure. According to [6], by moving content closer to the user, the quality of Internet content delivery (e.g., reduced latency and increased throughput) can be significantly improved. The storage capacity of the base station is limited, and a distributed content caching scheme can also be designed on the base station to minimize the content transmission delay. However, the cooperative cache of devices may sacrifice spectral efficiency. How to balance the relationship between content hit rate and spectral efficiency has not been well studied. Therefore, designing content caching schemes through content popularity and user preferences has also become a direction for researchers to explore the limitations of mobile edge cache size [7].

However, when we consider caching content, it is impossible to store all network content indefinitely due to the base station's storage capacity limitation or edge device. When the content base station requested by the user is not stored, it needs to request content from the remote data center, so that plays a role in reducing the delay. Secondly, although the edge cache can provide users with their requested content, as the number of portable devices increases, the user's mobility makes the content delivery process extremely complicated, which seriously affects the caching strategy [8–10]. In addition, the caching strategy designed by content popularity and user preferences, once the trend of data changes over time, we can not meet the needs of all users. Moreover, popular content may be popular for some people or popular content is not popular and cannot be captured in time. Therefore, for some users with special needs, relying only on the cache of popular content cannot meet the needs of these users. In this case, prefetching content through the backhaul will increase the burden on the network, causing network backhaul congestion and performance degradation. In the era of rapid data traffic, good strategies are needed to address user network requirements, reduce network latency, and improve the user experience. Therefore, to improve users' overall satisfaction, we need to give cache resources to users with niche needs and design a new content delivery method to break through the bottleneck of the existing transmission methods.

In order to reduce the delay as much as possible, the base station can only store the currently popular content, because most of the network traffic is composed of popular content. However, increasing the popularity hit rate still faces bottlenecks, as popular content is popular among most people, and the demand for 20%–40% of niche users (A small group with special hobbies) is still not met [11, 12]. In order to meet the needs of more users, the cache hit rate is increased while reducing the backhaul burden of network prefetching when meeting niche users. We have proposed the MPJUP scheme. We divide the cache capacity into a cache of popular content and a cache of prefetched content to meet the needs of niche users. The cache of prefetched data can provide priority services for single or multiple users. We predict the user's next location through the mobility of the user, and then the nearby SBS obtains the data predicted by the user through cache prefetching, reduces the delay of the user to obtain data, and improves the user experience. In addition, we have optimized the way to prefetch data. In order to reduce backhaul congestion, we consider the prefetching method of users participating in data submission. By comparing the cost of backhaul and prefetching data by the user, weigh the relationship between the number of task cooperation nodes and the message hit rate, find a suitable way to prefetch data, and reduce the cost of data prefetching. The main contributions of this paper are as follows:

- We use cached allocations to consider both popular content-based caching and caching based on niche user requirements. Furthermore, the device that prefetches data is judged by the user's mobility prediction to provide services for users with specific needs.
- We propose a prefetching method for joint user participation. By comparing the two ways of prefetching data, find the least costly way to get the data. Due to the prediction of user mobility and user cooperation in data prefetching, we can capture the user's dynamics in a short time, and adjust the cache content replacement strategy in time to reduce the backhaul load.
- When we propose that users cooperate to participate in data prefetching, we need to find the users who own the data and filter the users who upload the data to reduce the payment cost of users participating in cooperation and the cost of uploading data by users. Experiments show that the deviation of the content's popularity can better reflect the gain brought by our model.

2 Related work

In response to the explosive growth of mobile network traffic, the academic community has made many efforts in mobile edge caching. In the 5G network form, the base station construction is relatively dense, and deploying the cache at the base station is an important implementation of the mobile edge cache. Deploying a cache at the base station can greatly ease the pressure on the backhaul link and the mobile core network and reduce network latency. Base station cooperative caching [13] has aroused widespread concern in the academic community. By enhancing the content caching strategy and cooperation between base stations, content distribution can be further improved and the utilization of cache resources can be improved. By caching the content with the SBS closest to the user, when the user requests the content, the user can be served through the local cache to improve the request responsiveness. In this way, congestion of the backhaul link during peak hours of data requests can be avoided, and content delay can be reduced.

In actual situations, the capacity of the base station is limited. In order to effectively utilize the limited buffer size, the popular content can be cached in the base station [14, 15] to improve the quality of service of the mobile user. Since the terminal requests the popular content more frequently and places the popular content closer to the terminal user, the service delay and traffic load of the core network can be directly reduced, and the network congestion problem is indirectly solved. When a user requests large amounts of data such as video, the limited cache storage of the edge cache server cannot provide good services for it. Therefore, selecting the optimal cache content for the edge cache server can effectively reduce the base station service load and improve the user experience. In response to this problem, [16] studied the QoE-driven mobile edge cache optimization problem of dynamic adaptive video streams, and considered the coordination between distributed edge servers. The mobile edge server cache can smooth the change of time-domain traffic and reduce the base station's service load in data transmission.

Due to the cache capacity limitations of mobile edge servers, only some of the most popular content can be stored. However, the content popularity of locally cached content fluctuates over time, and the number of requests for each popular content may have a high degree of spatiotemporal variation. Moreover, the user's interest will also change with time. The existing edge caching strategy designed according to content popularity has certain challenges in the dynamic mobile network environment. A context-aware active caching scheme is proposed in [17]. They think that what the user likes may depend on the context of the user, and the algorithm updates the cached content by periodically observing the context information of the connected user. Learn the popularity of the context of the content online and use it to determine cache replacement decisions.

In addition, under the user-centric emerging network architecture, in order to improve the quality of user experience, each user can form an SBS cluster service [18, 19]. In this scenario, different users can store different content in a cooperative manner. When there is a user requesting content, the target content can be found from the local cache of the cluster SBSs with higher probability, and multiple base stations are allowed to provide services for the user. Optimize the content placement and cluster size based on random information such as network topology, traffic distribution, channel quality, and file popularity to optimize the cooperative edge cache [20]. The emerging layered network architecture enables us to increase the performance of content caching by opportunistically leveraging cloud-centric caching and edge-centric caching. A hybrid content caching scheme is proposed in [21] for this problem. It does not need to understand the content popularity and optimizes the content cache location to maximize the average requested content data rate.

In the cache policy design of edge devices in heterogeneous cellular networks, [22] studied the joint design and optimization of cache and user association strategies, and constructs a joint optimization problem by considering wireless channel quality and limited backhaul conditions to reduce backhaul delay. [15] proposes a collaborative content caching and delivery strategy that utilizes caching of popular content items at micro base stations (FBS) and mobile devices. Reduce the expensive transmission from MBS to mobile devices by considering the local cache construction joint optimization problem solving of the two devices. Similarly, the above two methods convert the content caching problem when considering the optimization problem, in order to reduce the backhaul burden of the requested content.

Based on the discussion of these methods, most of the problems in improving the content hit rate are by identifying popular content and then caching it. The end result of the method of continuously improving the popularity cache can only meet users who have demand for popular content, and these users cannot represent the preferences of all users. These methods improve network performance in terms of transmission delay, content hit rate or backhaul delay by utilizing various different information in the network. However, these methods are challenging in practice as user requests change and network channels change. Moreover, due to the variability of popular content, it is difficult to capture the characteristics of the user's mobility and the like with time. Users move to different locations over time and require different caching devices to provide services. Service scheduling cache scheduling through clustered SBS may be more frequent, and it is difficult to effectively reduce network cost and improve user experience in consideration of spectrum efficiency issues. To further improve cache and prefetch efficiency, reduce request latency, and improve the overall user experience. We need more complex decision-making solutions to meet the needs of more users. We judge the device that prefetches data by predicting the user's movement position with respect to the time by the user's mobility. In order to reduce the backhaul load more effectively, we propose a new way to reduce the backhaul load. Reduce the number of content backhaul by synchronizing prefetch data between users and edge caching devices. In addition, our algorithms are not just designed for small-scale networks. For large-scale networks, the way we combine users to prefetch data is also effective.

3 System model and problem statements

In order to reduce data transmission delays and improve user service satisfaction, we propose to prefetch data based on user mobility and store it in the base station. We set up two ways to prefetch data and proposed a content storage strategy.

In the 5G network, in order to meet the growth of data transmission speed, more small base stations (SBSs) need to be built. The mobile edge caching scenario of user pre-fetched content is shown in Fig. 1. Therefore, a multi-layer cellular network composed of single MBS, m SBSs and n random distributed UEs is formed. The set of SBS is represented as$S = \{s_1, s_2, \cdots, s_m\}$. The set of UE is represented as $U = \{u_1, u_2, \cdots, u_n\}$.

We assume that user u is requesting a data and may need subsequent data after time t, such as video data, audio data, and so on. We can improve the response speed of the request by prefetching the data requested by the user into the SBS. We can divide the cache into two parts, one serving the popularity-based cache of most users, and the other serving the cache that satisfies some user requests (also known as niche requests), that is, the cache based on user satisfaction. The main idea of the popularity-based cache is to do the cache replacement policy according to the most frequently requested data. Based on the user's individual cache, we mainly consider the user's mobility prediction about time to prefetch the data.

3.1 Determine the target of prefetched data

We use the mark of the small base station (SBS) as the location representation of its coverage area, for example, s_1 indicates the area covered by SBS s_1. Although the coverage area will overlap, we believe that the user always requests data from the SBS closest to it. We record the track of user u as $R^u = \{R^u_1, \cdots, R^u_\alpha\}$, where $R^u_\alpha = \{s_i\ , \lambda_i\ , \mathrm{o}_i\}$ represents the αth record of user u in the track information. s_i indicates the area where the αth track information user is located, λ_i indicates the time to reach the s_i area, and o_i indicates the time to leave the s_i area.

We use $R^u_i = \{s_i\ , \lambda_i\ , \mathrm{o}_i\}$ to simulate the movement of user u and record them with numbers. Since the probability that user u moves from area s_i to area s_j is independent of the area s_k that passes. Therefore, the user's trajectory R^u is a standard discrete-time Markov chain. Where $\mathrm{o}_i - \lambda_i$ describes the dwell time of user u in the area s_i where the α track record is located. These random variables are independently and identically distributed, and the distribution does not change over time.

We simply predict the next location of the user's mobile record. It can be assumed that our current position is known to be the s_i area and the time λ_i entering the s_i area, moving to the s_j area within t time. We use the time homogeneous semi-Markov model to define the core content of the relevant probability model as follows:

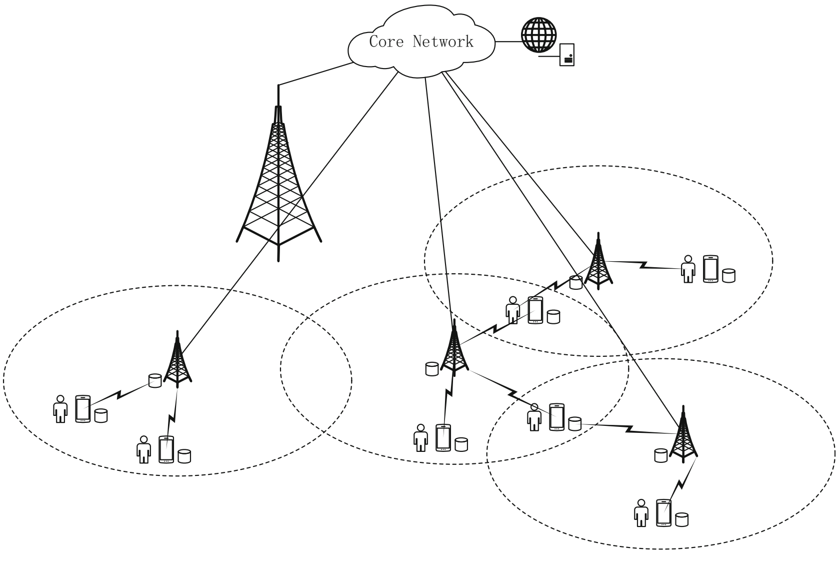


Fig. 1. the scenario of mobile edge caching with user prefetch.

$$\begin{aligned} B_{ij}^t &= P(R_{z+1}(t) = \lambda_i + t, R_{z+1}(s) = j, o_i - \lambda_i \le t | R_z(\mathrm{t}) = \lambda_i, R_z(s) = i) \\ &= P(R_{z+1}(s) = j, o_i - \lambda_i \le t | R_z(s) = i) \end{aligned} \tag{1}$$

Where B_{ij}^t represents the probability that user u will move from region s_i to regions_j after time t. It can be seen that the area s_j of the user after the time t depends on the user's previous movement record R_z, but has nothing to do with R_{z-1}.

The probability that user $u$moves from area s_i to area s_j before time t is as follows:

$$\begin{aligned} Q_{ij}^t &= P(o_i - \lambda_i \le t | R_z(s) = i, R_{z+1}(s) = j) \\ &= \sum_{\gamma=1}^{t} P(o_i - \lambda_i = \gamma | R_z(s) = i, R_{z+1}(s) = j) \end{aligned} \tag{2}$$

Therefore, we can get the probability that user u leaves area s_i before time t.

$$Q_i^t = P(o_i - \lambda_i \le t | R_z(s) = i) = \sum_{j=1, j \ne i}^{m} B_{ij}^t \tag{3}$$

$o_i - \lambda_i$ indicates the time when user u stays in area s_i. It can be seen that Q_i^t represents the residence time distribution of user u in area s_i, and is independent of the previous or subsequent track record.

To solve the user's probability of movement, we define the moving probability matrix of user u for the z-th record. Let us assume that the probability that user $u$moves from region s_i to region $s_j$is:

$$M_{ij} = P(R_{z+1}(s) = j | R_z(s) = i) = \frac{N(s_i, s_j)}{N(s_i)} \tag{4}$$

Where $N(s_i, s_j)$ is the number of transfers from the previous location of user u to s_i and the next location to s_j. $N(s_i)$ indicates the number of times the user u's previous location is in s_i and the next location is not in s_i.

The time homogeneous semi-Markov core part B_{ij}^t can be derived by (1), (2), (4):

$$\begin{aligned} & B_{ij}^t = P(R_{z+1}(s) = j, o_i - \lambda_i \le t | R_z(s) = i) \\ & = P(o_i - \lambda_i \le t | R_z(s) = i, R_{z+1}(s) = j) P(R_{z+1}(s) = j | R_z(s) = i) \\ & = Q_{ij}^t M_{ij} \end{aligned} \tag{5}$$

Through the above analysis, similarly, we can get another semi-Markov A_{ij}^t about time homogeneity. It represents the probability that user u is currently in region s_i and has moved to region s_j after time t. UnlikeB_{ij}^t, user u moves to s_j after time t. During this time, user u may have passed through zero or more regions other thans_j. Therefore, A_{ij}^t knows the current area's location and predicts the user's position in any time unit t over this period.

We consider the solution of A_{ij}^t in two cases. The first case considers that user u stays in area s_i until time unit$t$, and enters area s_j after t, without passing through other areas. In this case, the probability that the user does not leave the s_i area before t time is:

$$P(o_i - \lambda_i > t | R_z(s) = i) = 1 - Q_i^t \tag{6}$$

In the second case, we believe that user k does not only stay in area s_i before time unit t, but also passes through area s_k other than s_i, s_j and enters area s_i after time t. The probability in this case is:

$$\begin{aligned} & A_{ij}^t = P\Big(R_{z+1}(s) = j, R_{z+1}(t) = t | R_z(s) = i, \\ & R_z(t) = 0, R_z(s) = k, R_z(t) < t \cdots\Big) \\ & = \sum_{k=1}^{m} \sum_{\gamma=1}^{t} \left(B_{ik}^{\gamma} - B_{ik}^{\gamma-1}\right) A_{kj}^{t-\gamma} \end{aligned} \tag{7}$$

The above indicates the probability that the γ user has moved to the area s_k at a certain point in time t and moved to s_i after the time$t-\gamma$. The probability of accumulating the situation in which the movement area change may occur at these times is the probability ofA_{ij}^t in the second case. To sum up, it can be concluded that:

$$A_{ij}^t = \begin{cases} \sum_{k=1}^{M} \sum_{\gamma=1}^{t} \left(B_{ik}^{\gamma} - B_{ik}^{\gamma-1}\right) A_{kj}^{t-\gamma} & i \ne j \\ 1 - Q_i^t + \sum_{k=1, k \ne i}^{M} \sum_{\gamma=1}^{t} \left(B_{ik}^{\gamma} - B_{ik}^{\gamma-1}\right) A_{kj}^{t-\gamma} & i = j \end{cases} \tag{8}$$

Among them, we can know that the initial value of A_{ij}^0 is 0. When $i \ne j$, the initial value of A_{ij}^0 is 1. When $i = j$, we can use the result of $t = 0$ to calculate the value of A_{ij}at $t = 1$. Therefore, A_{ij} can be obtained by A_{ij}^0 and A_{ij}^{t-1}.

We can estimate the probability distribution of A_{ij}^t by calculating the probability value to predict the next location of user u after time t. The data is then prefetched by the SBS or edge device of the area.

3.2 The method of prefetching data

To achieve the information transmission speed of the 5G network, the number of SBS increases and users may have higher requirements on the response speed of the network [25–27]. In order to reduce data download delay and improve user experience satisfaction, many researchers proposed to cache data according to popularity in SBSs, but this can only meet the needs of some users for hot content. Taking into account the needs of niche users, we need SBS to have some room to serve other users who request content that is not currently popular.

When a user makes a data request to SBS, if the requested data has already been cached due to popularity, the user's needs are met locally. In the other case, if it is possible to predict the region in which the user will be in the next data request, the SBSs in that region can prefetch the data before the user arrives. It also allows users to access content directly from the edge of the network, reducing latency. We consider that SBSs has two ways of prefetching data. One common way is to obtain the content requested by the user through the managed server or CDN when the content requested by the user is not cached due to popularity in SBS. The other is that we think we can get it from the user who requested it and still has it. Next, we analyze the two cases:

Assume that the SBS cache capacity for prefetching data for the user is E. The cache content isς, the content size isC_ς, V_ς is the transmission rate of the content obtained from the remote, and the cost of prefetching the content from the remote server is:

$$\begin{aligned} &L_S = \frac{C_\varsigma}{V_\varsigma} + T_h + T_c \\ &s.t.\quad \frac{C_\varsigma}{V_\varsigma} \le t \end{aligned} \tag{9}$$

Where T_h represents the occupied network throughput and T_c represents the data transmission cost.

In the second case, we consider getting data from a node that has the content of the user's request. However, we do not have to send confirmation messages to all nodes that request this contentς, which increases the response's burden. We believe that the closer the request time is, the more likely the user is to retain the content. Therefore, assuming that the node has requested the data before timeσ, the probability that the data is still retained is defined as:

$$P_h = \begin{cases} \sum\limits_{k=1}^{x} \left(1 - \frac{\sigma_k}{t}\right) \theta_k & x \neq 0 \\ 0 & x = 0 \end{cases} \tag{10}$$

$$\theta_k = \begin{cases} 1 & k = x \\ \theta_{k+1} - \frac{1}{x} & 1 \le k \le x \end{cases} \tag{11}$$

The user requests x times of content ς in the T period, and σ_k represents the time between the kth request and the present. We sorted by the size of P_h and sent messages to nodes with a size of P_h greater than 0.5 to confirm whether it retains content ς and the size of the retained content $C_\varsigma^{'}$.We represent the set of users whose feedback has content ς and content size $C_\varsigma^{'}$ as H.

We consider that the content size $C_\varsigma^{'}$ retained by the user may affect the cost of prefetching content for the selected node. Because the rest of the content C_ς-$C_\varsigma^{'}$ still needs to be fetched from the remote server. First, we consider the cost of uploading data. There are two types of users who use mobile data traffic: those who need to pay for the use of data, which we call U_c, and those who use monthly data with unlimited monthly data, which we call U_f. Assuming that user U_f in user set H has a monthly traffic packet, the user directly uploads the content ς it owns, and if only part of the content$C_\varsigma^{'}$, the rest needs to be obtained remotely. Then the cost of prefetching data from the node is the cost of paying the user to participate in cooperation and the cost of uploading data. If user set H does not contain type U_f users, in order to save costs, we need to send data to type U_f users for upload. Therefore, selecting users with high probability of meeting type U_f users to participate in cooperative forwarding can reduce the cost of direct uploading data by type U_c users. Since the user needs to upload data before time t, it may not meet type U_f users within time t. Therefore, it may need other users with a higher probability of meeting type U_f users to cooperate in forwarding this data. However, such increase in users will not only improve the transmission success rate, but also increase the cooperation cost and transmission cost. Balancing the relationship between the number of cooperative users and the success rate of transmission is necessary. We propose a way to filter users to solve this problem.

In order to calculate the probability of encounter between a U_c-type user and a U_f-type user before time t, we consider the user as a node carrying a mobile device. We define that node u_a maintains its encounter status table entry information locally $E_a = \{<u_b, t_{ab}> \ldots\}$, u_a, $u_b \in U$. E_{ab}^k represents the kth encounter record of node u_a and node u_b. u_b is the encounter node in this record, and t_{ab} is the encounter time of nodes u_a and u_b. We randomly select the l segments in the time period T, the time of each segment is t, and the encounter relationship of the defined nodes is:

$$\kappa_{ab} = \frac{1}{l} \sum_{i=1}^{l} \sqrt{\frac{n_{ab}^{t_i}}{n_a^{t_i}} \cdot \frac{n_{ba}^{t_i}}{n_b^{t_i}}} \tag{12}$$

Where $n_{ab}^{t_i}$ is the number of encounters between nodes u_a and u_b of period$i$in time t, $n_{ab}^{t_i} = n_{ab}^{t_i}$.

The research shows that the encounter interval time between nodes obeys the exponential distribution [23], then the probability that the node encounters the U_f-type node u_b before the target node reaches region S_j at t is:

$$P_{ab}^t = 1 - e^{-\tau_{ab} t} \tag{13}$$

Where τ_{ab} represents the frequency of encounters between nodes u_a and u_b, we define:

$$\tau_{ab} = \frac{1}{\Delta\bar{t}_{ab}} \tag{14}$$

Where $\Delta\bar{t}_{ab}$ is the average of the encounter intervals of nodes u_a and u_b in the random ω-segment time t, defined as follows:

$$\Delta\bar{t}_{ab} = \frac{1}{l}\sum_{i=1}^{l}\frac{1}{n_{ab}^{t_i}}\sum_{k=0}^{n_{ab}^{t_i}}\left(t_{ab}^{k+1}-t_{ab}^{k}\right) \tag{15}$$

Where, t_{ab}^{k} represents the time when node u_a and u_b meet for the kth in the randomly selected i-segment t time, and the initial time t_{ab}^{0} is set to 0 in time t. According to the first three formulas (13), (14) and (15), the probability of any node u_a meeting other nodes in time t can be obtained.

$$\begin{aligned} P_{ab}^{t} &= 1-e^{\tau_{ab}t} \\ &= 1-\exp\left[-\omega(t-t_0)/\sum_{i=1}^{w}\frac{1}{n_{ab}^{t_i}}\sum_{k=0}^{n_{ab}^{t_i}}\left(t_{ab}^{k+1}-t_{ab}^{k}\right)\right] \end{aligned} \tag{16}$$

Where, t_0 represents the elapsed time since the initial state started. Therefore, before the time t when the target user reaches the region, the probability that user u_a meets type U_f user is:

$$P_{aU_f} = \sqrt[|U_f|]{\prod_{b\in U_f}\kappa_{ab}P_{ab}^{t}} \tag{17}$$

The cooperation node selection is shown in algorithm 1, calculate the probability of encounter according to algorithm steps 2–9. In order to balance the high cost caused by the excessive number of cooperative forwarding nodes, we set that users can only forward when they encounter a node with a higher probability of meeting the target node. And only send them to the node with the highest probability that they currently encounter, as shown in steps 16–20 of the algorithm. If the current highest is itself, the data is not sent, as shown in steps 14–15.

It is assumed that the payment cost for users to find cooperative nodes is β, and when users fail to find U_f-type users before the t time, the upload cost with traffic is $\mu C_\varsigma'$. Then we can obtain the upload cost of prefetch data using node method as follows:

$$\begin{aligned} L_N^u &= P_{aU_f}\cdot\beta + \sum_{h\in H}P_{hU_f}\beta + (|\Phi|-1)\beta \\ &\quad + \left(1-P_{aU_f}\right)\cdot\mu C_\varsigma' \end{aligned} \tag{18}$$

Where Φ is the set of cooperative nodes that the user is looking for, and $|\Phi|$ is the number of nodes in the set.

Therefore, we can get the cost of prefetching data through the node:

$$L_N = L_N^u + L_N^s = L_N^u + \frac{C_\varsigma - C_\varsigma'}{C_\varsigma}\cdot L_S \tag{19}$$

We can determine how the data is prefetched by comparing the size of L_N and L_S. If $L_N < L_S$, the data is prefetched through the node, otherwise the data is prefetched through the remote server.

3.3 Determines whether to prefetch the data

We determine whether to prefetch the data by the user's willingness to get the data for the faster payment y_ς. The specific optimization function is as follows:

$$G = \max_{\eta_\varsigma}\sum_{\varsigma\in S}\eta_\varsigma\left(\mu y_\varsigma - \min\{L_N, L_S\}\right) \tag{20}$$

Where η_ς indicates whether the data ς is prefetched, μ is the number of users requesting content ς. S is the data set requested by the user at the current time and the content is not in the popular cache.

3.4 Node filtering algorithm for cooperative prefetching data

Algorithm 1: Cooperative prefetching data of node filtering algorithm.

4 Simulation results

In this section, we validate our proposed method's performance based on system-level simulations, analyze the proposed method by numerical results, and study the value of the way we prefetch data through the user. The YouTube-8 M we use is a large tagged video dataset consisting of more than 6 million YouTube video IDs. According to public video data, popular videos account for a large percentage of clicks, while the least popular ones are rarely accessed. We understand that such a result indicates that the video distribution is a "long tail" distribution. Therefore, for content-based popularity-based caching strategies, we use Zipf to simulate the popularity distribution of content. In addition, we use the vehicle's real trajectory dataset for mobility prediction, which simulates large-scale vehicle movement trajectories in urban areas in detail, based on data provided by the TAPASCologne project. We judge the accuracy of the prediction by considering the K positions of the predicted next location. In the simulation process, we assume that the user is always connected to the nearest base station,

```
Input: $E_a$ Encounter status table record, $T$ period , $l$ segment with time t, $U_f$ A collection of users with unlimited monthly traffic, $U_c$ A collection of users who pay for traffic, $H$ The set of users who own content $\varsigma$
Output: $Y$ Cooperative node set
If $u_a \in \{U_f \cap H\}$ Then $u_a$ upload content $C'_\varsigma$;
Else If $u_a \in \{U_c \cap H\}$ Then
  Calculate $n_{ab}^{t_i}, n_a^{t_i}, n_b^{t_i}$ based on the record $E_a$ and $1 \le i \le l$
  $\kappa_{ab} \leftarrow getMeetCorrelation(n_{ab}^{t_i}, n_a^{t_i}, n_b^{t_i})$ // Node encounter relationship
  $\Delta\bar{t}_{ab} \leftarrow getIntervalMean(t_{ab}, n_{ab}^{t_i}, l)$ // The average of the encounter intervals of the nodes
  $\tau_{ab} \leftarrow getMeetFre(\Delta\bar{t}_{ab})$ //Node encounter frequency
  $P_{ab}^{t} \leftarrow getMeetPro(\tau_{ab})$ // Probability of encountering a $U_f$-type node before time t
  For each $a \in H$ , $b \in U_f$ do
    Calculate $P_{aU_f}$ according to $\kappa_{ab}$, $P_{ab}^{t}$ and formula(17)
  Let $x = P_{aU_f}$ , $Y = u_a$ , $a = \max_{i}\{P_{iU_f}\}$ , $i \in H$
  For k=1 to t do
    Let $x_k = x$
    For each $u \in Y$ do
      Let $X_k$ is the set of nodes that u encounters at k
      If $u_i \in X_k \cap U_f$ Then $u$ sends content $C'_\varsigma$ to $u_i$ and $u_i$ upload content $C'_\varsigma$;
      Else For each $u_i \in X_k$ do
          If $P_{u_iU_f} > x_k$
            $x_k = P_{u_iU_f}$
        End For
        $u$ sends content $C'_\varsigma$ to $u_i$ and asks it to forward it cooperatively , $Y.add(u_i)$
        $x = x_k$
  End For
End If
```

and also requests to download data from the nearest base station. Each base station has 10 GB of content storage space, and the average link bandwidth distribution at a base station density of 250 m, 200 m, and 150 m is 2 Mbps, 3 Mbps, and 4 Mbps. We assume that there are 200,000 files in the remote content server, each with a file size of 200 MB. Moreover, the content popularity distribution follows the Zipf distribution [24]. Defined as:

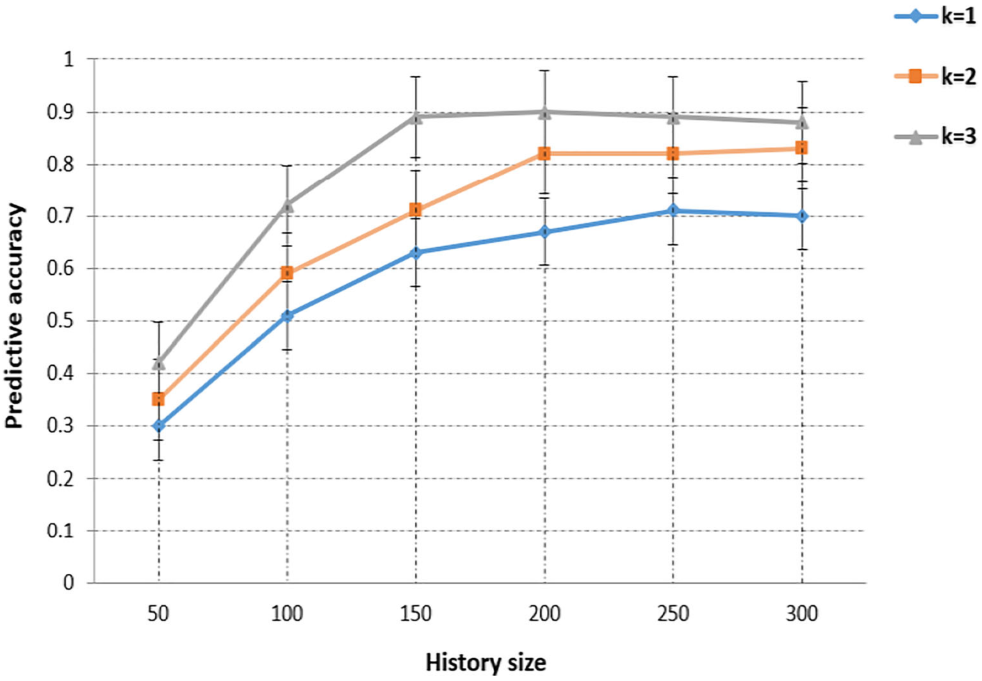
$$q_\tau = \frac{1/\tau^\delta}{\sum_{k=1}^{S} 1/k^\delta} \tag{21}$$

Where $\delta \geq 0$represents the skewness of the popularity distribution, and $\delta$is larger to indicate a more centralized file request. Therefore, we represent the diversity of content by changing the values of different popularity distributions.

4.1 Mobility prediction accuracy

The first step in our model is to predict the mobility of the user. We predict K-locations (or connected base stations) that users may arrive after time t by predicting time-related mobility predictions. In order to achieve a user's mobility assessment, different historical data sizes and predicted the number of next locations k will have an impact on user mobility accuracy. Our results are shown in Fig. 2. Our spatial position prediction for the user shows that the larger the K number of the base station that is predicted to be connected next, the higher the accuracy of the prediction. The larger the number of historical samples, the higher the accuracy. However, their predictions will eventually approach equilibrium, as more historical samples can only have a minor impact on recent mobility changes. The same is true for the next number of positions, and predicting a low probability of reaching a location

Fig. 2 Historical sample size and mobility prediction accuracy



may only have a small effect on the outcome. Therefore, we can achieve relatively high precision with a small k value.

4.2 Prefetch data evaluation

4.2.1 Cache hit ratio evaluation

Through model analysis, we can know that our method is to prefetch data based on the satisfaction of more users. Our method's total cache space allocates a certain percentage of the user experience based prefetch data space and the popularity cache based space, while the other two strategies do not allocate space. We can see from Fig. 3 that as the Zipf index increases, the hit rate of the three strategies increases, and the hit rate converges when the Zipf index increases to a certain extent. We can see that when the Zipf value is small, the popularity distribution of the content is not obvious. At this time, the pre-fetching of the gain based on the user experience is more advantageous than the cache based on the

Fig. 3 Zipf distribution and cache hit ratio

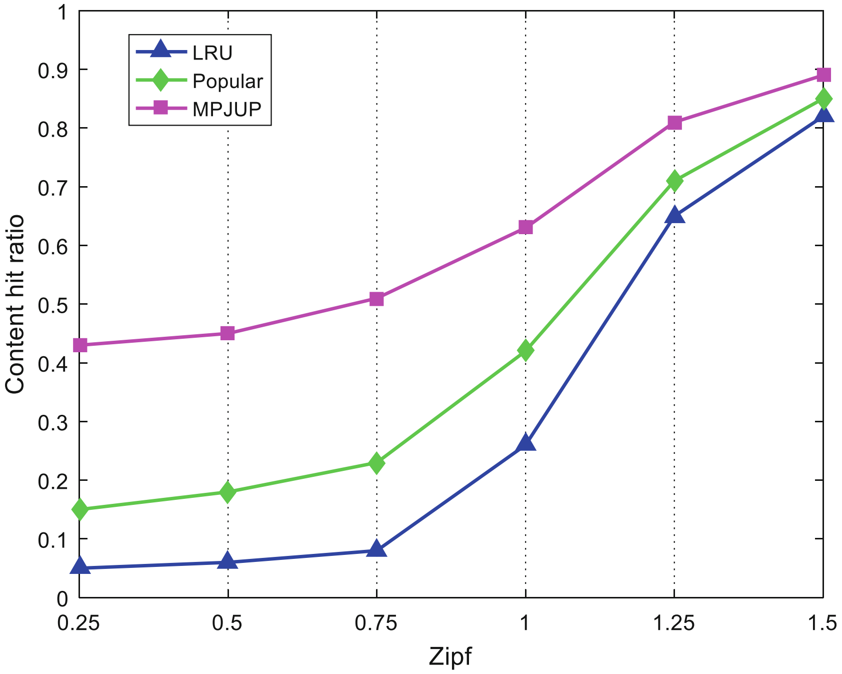
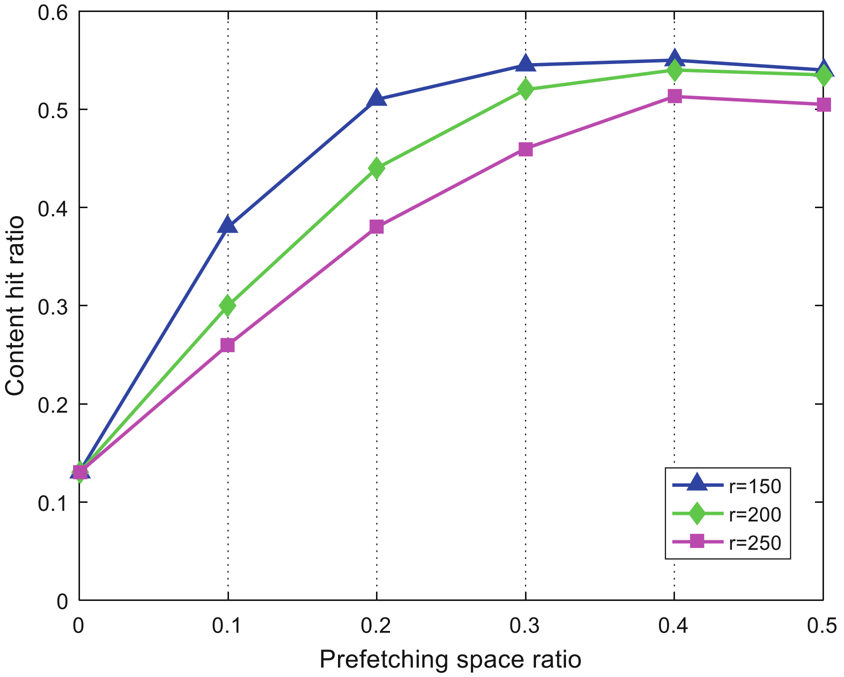


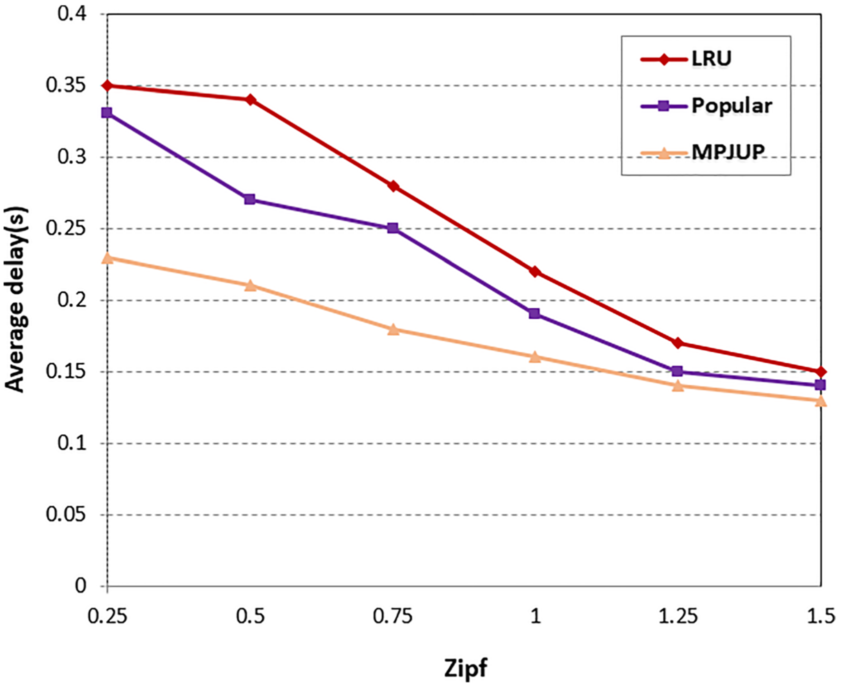
Fig. 4 Prefetch space ratio and cache hit ratio



popularity. As the Zipf value increases, the distribution of popularity becomes more and more clear. At this time, the advantage of prefetching data is reduced, and the difference between the three strategies becomes small. However, our proposed strategy is always superior to other strategies in terms of hit rate, because our strategy fully considers users with niche needs, using the cached content of surrounding users as a supplement to the base station cache, improving the hit rate of user requests.

We allocate a certain amount of prefetched data space to the cache space in the scheme. We demonstrate the merits of our approach by analyzing the relationship between the ratio of pre-fetch space and the hit rate. And we divide our strategy's SBS intensity into three values, and the distances are 150 m, 200 m, and 250 m, respectively, to compare their differences. We set the Zipf index parameter value to 0.75 according to the normal value of the content hit ratio. As shown in Fig. 4, as the prefetch space

Fig. 5 Zipf distribution and average latency



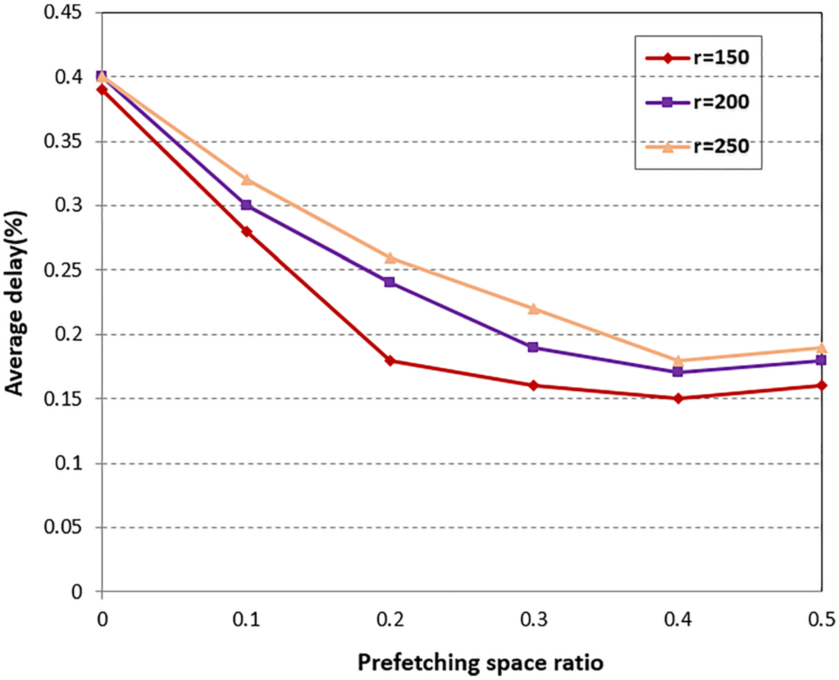


Fig. 6 Prefetch space ratio and average latency

accounts for the increase of the total cache ratio, the hit rate of our strategy is increasing. When the proportional index is increased to a certain extent, the hit rate converges and declines. However, when the scale values are small and large, their differences are not significant. However, due to the difference in density, the base station's workload is different, so the value of the optimal prefetch space is different. We can guess that the more base stations will reduce the workload of a single base station. Therefore, the smaller the base station density, the less prefetch space is needed to achieve better performance. However, we can also see from the figure that the cost of less prefetch space is much higher than that of a larger prefetch space. As described in Fig. 3, when the Zipf value is 0.75, prefetching user request data can bring greater gain to the cache hit ratio.

4.2.2 average delay

In order to evaluate the performance of our niche demand prefetch method based on user experience gain in the case of content popularity distribution changes. Fig. 5 depicts the effect of the average latency on the three strategies under a change in content popularity distribution. As the zipf index increases, the average delay of the three schemes gradually decreases. Because the popularity of the user's requested content is more concentrated, the probability of the user's desired content being cached in the base station becomes higher, that is, the requested hit rate is increased

When the Zipf value is large, the impact of the three strategies on the hit rate is reduced. The increase in the hit rate allows the base station to have more content to directly transfer content from the local buffer to the user without taking data from the remote data center. The average delay in our proposed scheme is significantly lower than other schemes when the content popularity distribution is less obvious. As in the previous analysis, when the Zipf value is small, the prefetched data has a greater impact on the hit rate. Because our solution can prefetch data for users, we can extract the niche content directly from nearby users and reduce the backhaul delay when the user requests it because the content has been cached locally.

In order to meet the needs of some niche users, our solution allocates a certain proportion of cache space to prefetch data for users to reduce latency. We analyze the relationship between the ratio of prefetch space and the average delay and compare the results by the degree of influence in different SBS intensities. As can be seen from Fig. 6, as the prefetch buffer space increases, the average latency of user requested data decreases continuously, and the minimum delay is achieved at different SBS densities at different prefetch space ratios. When prefetching space after the size exceeds this ratio, the average delay starts to rise again. This is because the prefetch buffer can only be used as a supplement to the user's needs and cannot completely replace the popularity-based cache. Especially when the SBS density is relatively small, it is of little value for the niche users to take up too much cache space. In other words, we can not harm the interests of the public when solving the problem of niche demand. However, prefetching the buffer to set the appropriate weight is able to substantially reduce the average delay.

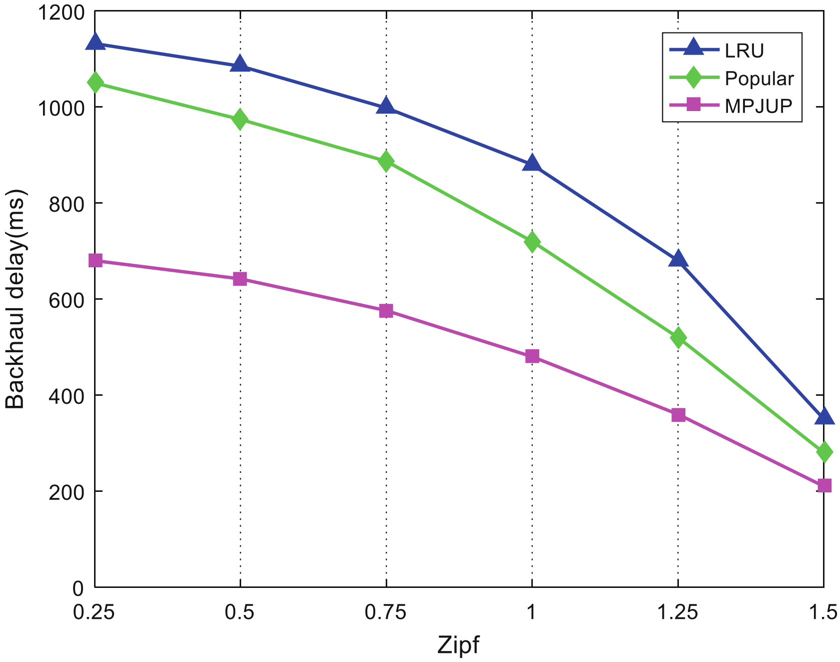


Fig. 7 Zipf distribution and backhual delays

4.2.3 Prefetch Data Cost assessment

In order to reflect the advantages of our joint user cooperation method of prefetching data, we compare the load of the backhaul link of the three strategies when the popularity distribution changes. As can be seen from the figure, as the Zipf index increases, the backhaul link load becomes smaller. We can see from the previous relationship between Zipf and cache hit ratio in Fig. 7 that as the Zipf value increases, the cache hit rate increases. Therefore, as the content requested by the user is more concentrated, the cache hit rate is higher, so that the number of times the content needs to be acquired from the remote data center because the requested content is not cached can be greatly reduced, and the data can be locally obtained, thereby reducing the backhaul load. However, our strategy can get better results even when the Zipf value is low. Because we have frequent space updates for prefetching data, we have further reduced the number of backhaul and backhaul block sizes by means of user cooperation and prefetching data. Our goal is not only to reduce data transfer latency but also to minimize backhaul load.

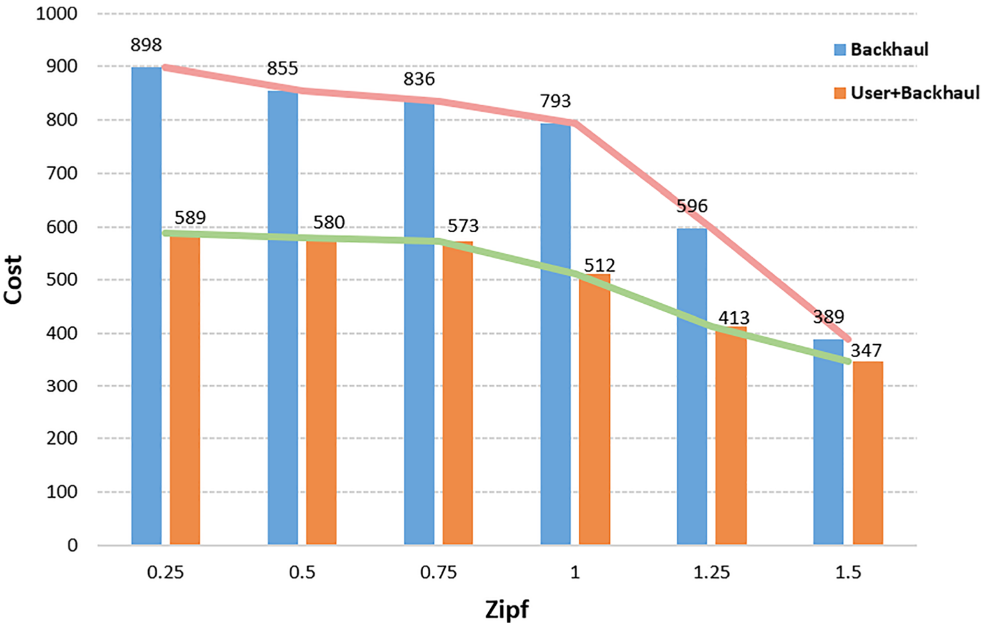


Fig. 8. zipf distribution and cache cost.

The model we designed takes the cost for the base station to fetch data from the remote data center. Of course, it takes the cost to fetch data from users near the base station, but the cost of fetching data from the user is mostly lower than the cost of fetching data from a remote data center. At the same time, we do not consider the transmission cost of the base station to provide data to the user locally. Here, we compare the base station's cost in the different processing situations when the user requests content is not cached. As the Zipf index increases, the cost of providing services to users by the base station is steadily decreasing. As the cache hit rate increases, the amount of content that can be sent directly from the local cache to the user increases. As we can see from Fig. 8, the way of pre-fetching through user collaboration is always lower than the cost of direct remote prefetching of data. Because we compare the two methods of prefetching data by predicting the cost and choose a method that predicts the lower cost to perform prefetching. Moreover, the overall data prefetching cost can be further reduced by acquiring content requested by other users from users in the vicinity of the base station.

5 Conclusions and future work

We propose a new model based on user mobility and user experience gain cache, and propose a method for joint users to perform mobility prefetching. Our solution not only considers the needs of the mass users but also considers the users for specific needs and meets the needs of niche users. We can dynamically adjust the prefetched content by prefetching data and reduce the backhaul load caused by the continuous update of the prefetch space through joint user prefetching. Our approach is primarily a caching strategy for less popular content, as this part of the content is also concerned by niche users. Although these contents are not of concern to the public, the niche benefits they can bring are not to be underestimated. The verification shows that the performance of our method in terms of hit rate and average latency is improved due to the targeted caching strategy for specific users. And the method of caching based on the user experience gain also satisfies the needs of more users, and the QoE of more users is improved to some extent. In the future research direction, we will study more complex mobility prediction algorithms and more accurate predictions about user needs, and consider the method of user cooperative caching to reduce backhaul load and average latency further.

Acknowledgments This work was supported in The National Natural Science Foundation of China(61672540); Hunan Provincial Natural Science Foundation of China (2018JJ3299, 2018JJ3682). Fundamental Research Funds for the Central Universities of Central South University (2019zzts152).

References

1. Li, H., Ota, K., Dong, M.: Virtual Network Recognition and Optimization in SDN-enabled Cloud Environment. IEEE Transactions on Cloud Computing, 1–1 (2018). https://doi.org/10.1109/TCC.2018.2871118
2. Li H, Ota K, Dong M (2018) ECCN: orchestration of edge-centric computing and content-centric networking in the 5G radio access network. IEEE Wirel Commun 25(3):88–93. https://doi.org/10.1109/mwc.2018.1700315
3. Dong M, Ota K, Li H, Du S, Zhu H, Guo S (2014) Rendezvous: towards fast event detecting in wireless sensor and actor networks. Computing 96(10):995–1010. https://doi.org/10.1007/s00607-013-0364-7
4. Dong M, Ota K, Yang LT, Liu A, Guo M (2016) LSCD: A Low-Storage Clone Detection Protocol for Cyber- Physical Systems. IEEE Trans Comput-Aided Design Integrat Circuits Syst 35(5):712–723. https://doi.org/10.1109/TCAD.2016.2539327
5. Dong M, Ota K, Liu A, Guo M (2016) Joint optimization of lifetime and transport delay under reliability constraint wireless sensor networks. IEEE Trans Parallel Distribut Syst 27(1):225–236. https://doi.org/10.1109/TPDS.2015.2388482
6. Guan P, Wu J (2019) Effective data communication based on social Community in Social Opportunistic Networks. IEEE Access 7:12405–12414. https://doi.org/10.1109/ACCESS.2019.2893308
7. Wu J, Chen Z, Zhao M (2019) Weight distribution and community reconstitution based on communities communications in social opportunistic networks. Peer-to-Peer Networking Appl 12(1):158–166. https://doi.org/10.1007/s12083-018-0649-x
8. Wu J, Yu G, Guan P (2019) Interest characteristic probability predicted method in social opportunistic networks. IEEE Access 7:59002–59012. https://doi.org/10.1109/ACCESS.2019.2915359
9. Wu J, Chen Z (2018) Sensor communication area and node extend routing algorithm in opportunistic networks. Peer-to-Peer Networking Appl 11(1):90–100. https://doi.org/10.1007/s12083-016-0526-4
10. Wu J, Chen Z, Zhao M (2019) SECM: status estimation and cache management algorithm in opportunistic networks. J Supercomput 75(5):2629–2647. https://doi.org/10.1007/s11227-018-2675-0
11. Zhang D, Qiao Y, She L, Shen R, Ren J, Zhang Y (2019) Two time-scale resource Management for Green Internet of things networks. IEEE Internet Things J 6(1):545–556. https://doi.org/10.1109/JIOT.2018.2842766
12. Zhang D, Chen Z, Zhou H, Chen L, Shen X (2016) Energy-balanced cooperative transmission based on relay selection and power control in energy harvesting wireless sensor network. Comput Netw 104:189–197. https://doi.org/10.1016/j.comnet.2016.05.013
13. Shanmugam K, Golrezaei N, Dimakis AG, Molisch AF, Caire G (2013) FemtoCaching: wireless content delivery through distributed caching helpers. IEEE Trans Inf Theory 59(12):8402–8413. https://doi.org/10.1109/TIT.2013.2281606
14. Liu, J., Bai, B., Zhang, J., Letaief, K.B.: Content caching at the wireless network edge: A distributed algorithm via belief propagation. In: 2016 IEEE International Conference on Communications (ICC), 22–27 May, pp. 1–6 (2016)
15. Jiang W, Feng G, Qin S (2017) Optimal cooperative content caching and delivery policy for heterogeneous cellular networks. IEEE Trans Mob Comput 16(5):1382–1393. https://doi.org/10.1109/TMC.2016.2597851
16. Müller S, Atan O, Schaar MVD, Klein A (2017) Context-aware proactive content caching with service differentiation in wireless networks. IEEE Trans Wirel Commun 16(2):1024–1036. https://doi.org/10.1109/TWC.2016.2636139

17. Chen S, Qin F, Hu B, Li X, Chen Z (2016) User-centric ultra-dense networks for 5G: challenges, methodologies, and directions. IEEE Wirel Commun 23(2):78–85. https://doi.org/10.1109/MWC.2016.7462488
18. Li C, Toni L, Zou J, Xiong H, Frossard P (2018) QoE-driven Mobile edge caching placement for adaptive video streaming. IEEE Trans Multimed 20(4):965–984. https://doi.org/10.1109/TMM.2017.2757761
19. Bao, W., Liang, B.: Stochastic geometric analysis of handoffs in user-centric cooperative wireless networks. In: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 10–14 April, pp. 1–9 (2016)
20. Chen Z, Lee J, Quek TQS, Kountouris M (2017) Cooperative caching and transmission Design in Cluster-Centric Small Cell Networks. IEEE Trans Wirel Commun 16(5):3401–3415. https://doi.org/10.1109/TWC.2017.2682240
21. Kwak J, Kim Y, Le LB, Chong S (2018) Hybrid content caching in 5G wireless networks: cloud versus edge caching. IEEE Trans Wirel Commun 17(5):3030–3045. https://doi.org/10.1109/TWC.2018.2805893
22. Wang Y, Tao X, Zhang X, Mao G (2016) Joint caching placement and user Association for Minimizing User Download Delay. IEEE Access 4:8625–8633. https://doi.org/10.1109/ACCESS.2016.2633488
23. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Performance analysis of mobility-assisted routing. Paper presented at the proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, Florence, Italy
24. Tran TX, Le DV, Yue G, Pompili D (2018) Cooperative hierarchical caching and request scheduling in a cloud radio access network. IEEE Trans Mob Comput 17(12):2729–2743. https://doi.org/10.1109/TMC.2018.2818723
25. Zhang D, Chen Z, Awad MK, Zhang N, Zhou H, Shen XS (2016) Utility-optimal resource management and allocation algorithm for energy harvesting cognitive radio sensor networks. IEEE J Select Areas Commun 34(12):3552–3565. https://doi.org/10.1109/JSAC.2016.2611960
26. Zhang D, Chen Z, Ren J, Zhang N, Awad MK, Zhou H, Shen XS (2017) Energy-harvesting-aided Spectrum sensing and data transmission in heterogeneous cognitive radio sensor network. IEEE Trans Veh Technol 66(1):831–843. https://doi.org/10.1109/TVT.2016.2551721
27. Zhang, D., Tan, L., Ren, J., Awad, M.K., Zhang, S.,Zhang, Y., Wan, P.: Near-optimal and Truthful Online Auction for Computation Offloading in Green Edge-Computing Systems. IEEE Trans Mobile Comput, 1–1 (2019).https://doi.org/10.1109/TMC.2019.2901474

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

GengHua Yu is Ph.D Candidates in School of computer science and engineering, Central South University, Chang-sha, Hunan, P.R.China, in 2017. She is the 2017 outstanding graduate of Nanchang University. Her research interests include wireless communications and networking, medical informatics, big data research.

Jia Wu received the Ph.D. Degrees in software engineering Central South University, Changsha, Hunan, P.R.China, in 2016. He is associate professor in School of computer science and engineering, Central South University. Since 2010, he has been Algorithm engineer in IBM company in Seoul, Republic of Korea and in Shang-hai, P.R.China. He is a senior member of CCF(China Computer Federation), a member of IEEE and ACM. His research interests include wireless communications and networking, medical informatics, big data research, mobile health in network communication.