An Algorithm Based on Markov Chain to Improve Edge Cache Hit Ratio for Blockchain-Enabled IoT

Hongman Wang[1,2], Yingxue Li[1,*], Xiaoqi Zhao[1], Fangchun Yang[1]

[1] Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] Engineering Research Center of Information Network, Ministry of Education, Beijing 100876, China
* The corresponding author, email: silviali1225@163.com

Received: Nov. 26, 2019
Revised: Feb. 18, 2020
Editor: Wuhui Chen

Abstract: Reasonable allocation of storage and computing resources is the basis of building big data system. With the development of IoT (Internet of Things), more data will be brought. A three-layer architecture includes smart devices layer, edge cloud layer and blockchain-based distributed cloud layer. Blockchain is used in IoT for building a distributed decentralize P2P architecture to deal with the secure issue while edge computing deals with increasing volume of data. Edge caching is one of the important application scenarios. In order to allocate edge cache resources reasonably, to improve the quality of service and to reduce the waste of bandwidth resources, this paper proposes a content selection algorithm of edge cache nodes. The algorithm adopts markov chain model, improves the utilization of cache space and reduces the content transmission delay. The hierarchical caching strategy is adopted and the secondary cache stores slides of contents to expand the coverage of cached content and to reduce user waiting time. Regional node cooperation is adopted to expand the cache space and to support the regional preference of cache content. Compared with the classical substitution algorithm, simulation results show that the algorithm in this paper has higher cache hit ratio and higher space utilization.

Keywords: cache resource allocation; blockchain-enabled iot; edge computing; Markov chain; hierarchical caching technique

I. INTRODUCTION

As Internet of things (IoT) technology is developing rapidly, the number of IoT devices is explosion in our society which is predicted to be 25 billion by 2021[1, 2]. These distributed devices are producing a huge volume of data streams and transmitting the data to the remote cloud to deliver services. This leads to the secure, scalable, and efficient resource management issues for trusting third parties to secure privacy and sensitive data. To alleviate these problems, a distributed blockchain cloud architecture is proposed. With the blockchain technology, applications can be operated in distributed way instead of passing through a trusted intermediary[3]. With the edge computing technology, the limited computing resources of IoT devices which cannot support the growing demand of IoT data issue can be solved. Resources can be used more closer to the edge of IoT network, which can provide faster cloud services such as storage, computing and networking capabilities to end users[3].

However, some applications need local

storage for fast transmission and low response time[4, 5]. Considering the expensive transmission costs like bandwidth, time and energy, deploying content cache on the edge of the network has become the focus of the academia and industry, which can avoid deplication data transmission. With the rapid development of the content delivery network (CDN) and the information center network (ICN), content caching is introduced to utilize the storage capacity of edge nodes in blockchain-enabled IoT. Deploying the cache closer to the user's network edge can effectively relieve the pressure of backlink bandwidth, reduce repeated transmission and the network latency.

European telecommunications standards institute for the first time proposed in 2014, that MEC system allows equipment offload computing tasks to the network edge node such as base stations and wireless access points in order to improve the transmission speed and the quality of service. Thus the edge of computing is facing problems of node selection and resource allocation. The virtual resource allocation strategy including storage mentioned in [6] is also one of the problems in the future development of edge computing. Data caching, for example, is an important method for the mobile edge computing and caching technology. It is the trend of the current evolution of a variety of technology developments. Good selection strategy can reduce the service cost and improve the user experience. Figure 1 shows the distributed blockchain cloud ar-

In this work, we present a decentralized prediction market platform based on blockchain and masternode technologies. This paper proposes a content selection algorithm of edge cache nodes.

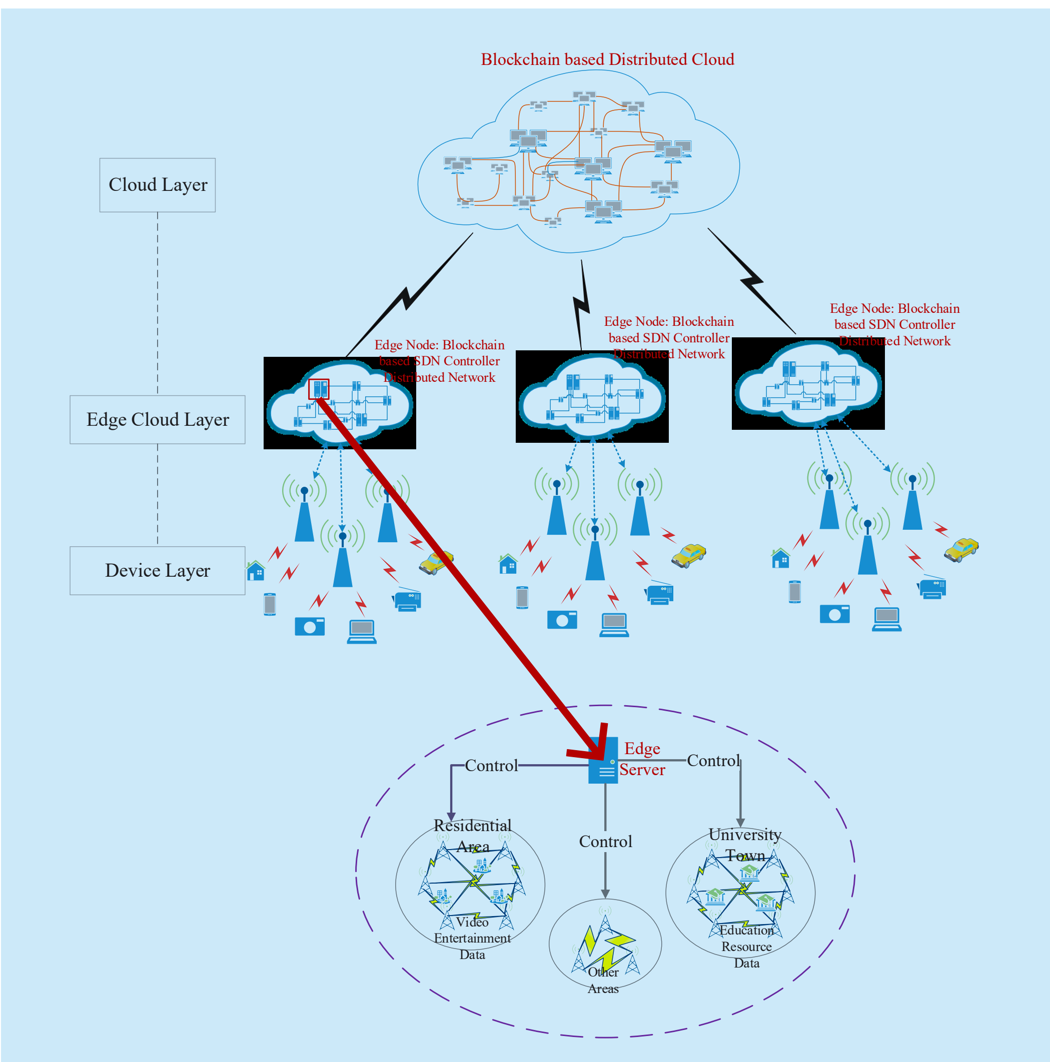


Fig. 1. *Distributed blockchain cloud architecture.*

chitecture combined the edge content cache scene.

This paper mainly focuses on content caching and a node selection algorithm for blockchain-enabled IoT is proposed, which provides a low latency content caching strategy for content providers in the edge cloud layer. So providers can provide users with a stable and efficient service to extract the content. The main contributions include the following:

- Proposing a node selection strategy of content deployment which aims to improve the quality of service and to reduce the waste of bandwidth resources.
- Discussing the impact of user mobility and Markov chain model is fused to the replacement strategy on the edge cache to improve the hit rate of cache files and to reduce the content transmission delay.
- Making each cache node work together on a regional scale to make the mobile network architecture flatter and stronger.

The remainder of this article is organized as follows. Section 2 presents the outcomes of blockchain-enabled IoT, edge computing and edge caching, traditional content distribution network. In section 3, an edge cache node content selection replacement algorithm is proposed followed by the evaluation of the proposed algorithm in section 4. Section 5 presents the conclusions and future work of this research.

II. Related Work

At present, the development of IoT creates a huge amount of content, which means the demand of bulky processing units, content caches and bandwidth provision[7]. In order to address the safety of data storage and sharing issues, blockchain technology is utilized to preserve the privacy of users[8]. To deal with real-time processing and feedbacking the storage of the content caches and in IoT environment[9], the edge computing and edge cache have caught the attention of the academia.

2.1 Blockchain-enabled IoT

Blockchain technology has attracted attention of researchers in establishing a secure, trusted and decentralized transport ecosystem which can sole data sharing issues. Kangs et al. developed a secure P2P data sharing system for vehicular data by vehicular blockchain, which performs distributed data storage and secure data sharing[8]. Sharma et al. used SDN and blockchain techniques to bring computing resources to the edge of the IoT network and allowed low latency access to huge volume of data in secure manner[3]. The novel blockchain-based distributed edge cloud architecture can provide high availability, real-time data delivery, scalability, security and low latency to deal with the explosion of data volume produced by smart devices. Zhenni Li et al. mainly focused on the computing resource trading issue for edge-cloud-assistd IoT under blockchain environment[10]. Jinliang Xu et al. focused on a crowd-intelligence ecosystem[11] and Hongxin Wei et al improve the efficiency of blockchain systems from the communication perspective[12]. However, few papers pay attention to content cache deployment issue under blockchain-based distributed edge cloud architecture, which can relieve the pressure of backline bandwidth and reduce repeated transmission and the network latency.

2.2 Content distribution network

CDN is short for content distribution network. It relies on the edge servers deployed everywhere, through load balancer, content distribution, scheduling and other functional modules of the central platform, so that users can get the required content from the nearest network. CDN can improve the user's access response speed and hit rate[13], and reduce the information transmission delay, alleviate the pressure of the return of the link bandwidth.

These CDN suppliers mainly rely on the IDC room rental equipment. But from the operator's point of view, there are three layers between mobile users and content, which are transmission network, core network and

CMNet network (China Mobile communication, for example). Signals are sent through PTN, RNC (MME), SGSN and GGSN, FW, ROUTER and other network elements. With the construction of 4G networks, this part of the load on the network traffic surges in[14]. In order to improve the user experience and to reduce the traffic pressure of the transmission network, sinking the edge content node to the core network or transport network will shorten the distance between the user and content, lower access latency and improve the stability of the access link. At the same time, due to the wide range of CDN service, far distance between nodes, long echo path, and long transmission time, the mobile Internet gateway is under pressure with higher cost of CDN construction and complex technology, and it is sensitive to the damage of single node.

2.3 Mobile edge computing

Mobile edge computing (hereinafter referred to as MEC) deploy computing power on the access network side by offloading computing from terminal to edge computing node[15], so that the cloud computing technology can also be used in the access network side to realize the unification and integration of communication and computation. Xiaoyu Qiu et al. proposed a new online deep reinforcement learning based online computation offloading algorithm which can maxmize long-term offloading performance and increase convergence rate[16]. In traditional LTE networks, the data surface function mainly concentrates on the LTE networks and the Internet on the boundary of P-GW, and requires that all the data flow must pass a P-GW user even for the same cell. This brings difficulties to deployment of the data flow between internal new content to the network application service. At the same time, excessive concentration of data surface functions also puts forward higher requirements for the performance of P-GW, which will easily lead to the bottleneck of network throughput of P-GW.

Through service localization, cache acceleration, local data diversion, flexible routing, computation offloading and other technologies, MEC technology can effectively reduce the demand for network return bandwidth and relieve the data transmission pressure of core network. MEC technology is an effective means of 5G network business applications to achieve near-user deployment and access, and has flexivle user interface. It can provide users with low latency and high bandwidth of transmission capacity, create virtual RAN LAN. The main functions of the 5G controller are deployed in the cloud of centralized control. MEC uses edge cache technology to support multimedia contents in 5G, which can reduce the latency effectively[17].

In [18], the authors deployed the CDN mechanism to the mobile platform by studying the storage capacity of mobile devices, and studied the corresponding content replication problem. In [19], the authors studied the energy efficiency of CDN system based on grid WIFI. And there is a shortcoming that the above works have not considered the control module on BSs.

In [20], the authors proposed a reactive caching strategy for BS caching system which can reduce the traffic on the return link. In [21], the authors suggested implementing caching on small cells to reduce traffic pressure on the return link. They studied the best content placement problem to minimize file download time. But there is also a disadvantage that the above research content is based on the assumption that BSs can cooperate in the whole mobile network without considering the mobility of users and they will frequently switch in and out. In [22], S.Wang proposed a service recommendation approach based on collaborative filtering and made QoS prediction based on user mobility. However, the mobile network is a large-scale system involving a large number of BSs. Collaborations across the network lead to complexity in content discovery, user request redirection, and load balancing. For making balance the workloads of edge servers and minimizing the access delay between the mobile user and edge server, in [23] they adopted mixed integer programming to find the optimal solution.

Therefore, in this paper, each cache node works together on a regional scale to make the mobile network architecture flatter and stronger. There are also some related studies which aim to improve the transmission efficiency, without considering the cache utilization efficiency and the cache content hit ratio when the capacity is short. Since the transmission efficiency has been substantially improved when the node drops to BS, the problem of limited cache space is highlighted.

III. Edge Cache Node Content Selection Replacement Algorithm

3.1 Algorithm localization advantage

The edge computing node with storage capacity is replaced by the traditional CDN node, which is closer to the wireless access network and has faster response speed. However, it brings the problem of deploying more nodes and more repeated content deployments. Therefore, this paper focuses on the node selection strategy of content deployment which aims to reduce the computing resource.

When the cache node sinks to the wireless access network side, the coverage range and the capacity of the cache node decrease, and the distance between nodes shrinks. Repeated deployment of content to the cache node without strategy will waste resources and increase the cost. Considering the cooperative deployment of nodes within a certain range the content of the cooperative nodes within the scope is not repeated. When the users in the cooperative area get the content of the cache in the cooperative node, the latency of the information is much shorter than the original storage server of the request content provider.

Cache content in cooperative area is more biased towards local users' preference for resources. Therefore in different regions, the cache content will inevitably show regional differences according to local user requests, which is more in line with the expectation of efficient use of cache resources.

At the same time, for the user request to roam to the local, if the content of the request is individualized and different from the local user's request, the user is at a disadvantage when accumulate and calculate the probability of content replacement. Therefore, the accidental personalized request does not cause the huge fluctuation of the content of the cache, that is, the algorithm has strong robustness, and can better serve the local resident users.

This article involves the basic process of user's request for resources: when the user requests content x, the algorithm first retrieves the cache for the resource x from the cooperative edge cache node in the area of the user. If hit, it returns the hit result directly, otherwise the content provider is requested to deploy the CDN node or the source server, until the content is returned.

3.2 User mobility management reduces cold boot problems

When the cache node sinks to the wireless access network side, the coverage range is greatly reduced, and mobile users will frequently cut in and out. Users' short stay and fast switching will bring great interference and resource waste to the edge cache system. At the same time, the addition of new users will cause cold boot problems frequently, and reduce the cache hit ratio. Therefore, user mobility management is necessary.

Just like the common switching task of mobile users, when mobile users are continuously requesting resources while moving, the edge cache node will be switched. So predicting the user's mobile trajectory and transferring data to the next cache node in advance can improve the user's access speed. This can relieve the interference of mobile user to local cache data and the cold boot problem to a certain extent.

Predicting the location of mobile users is of great importance in many telecommunications applications, including mobile access control, resource management, and content migration. Antonescu[24] and others migrate users to service prevention and extension, and perform distributed cloud service choreography on the basis of predicting user mobility: the system predicts that users will move to away from each data center (data center, DC) area to allocate more or less capital. However, it does not consider to migrate services to other locations.

In this direction, Follow-Me Cloud service migration strategy is an important concept, which puts forward by Taleb[25]. it assumes that the mobile network is supported by the great core DC, many small DCs are deployed in network edge geographically, also suggested that cloud services should be deployed in small DCs. Based on Markov, Taleb and Ksentini[26, 27], etc try to keep cloud service on the optimal DC according to users' mobility, interests and other network-related factors, but does not considering multiple destinations of users' paths. Andre S et al[28] propose the end-to-end m-FMC model, which supports content migration and adopts advanced strategies to consider the optimal content migration predicted by user and group migration. Unlike other FMC methods, it allows content to be migrated at different cache levels depending on how close the user is. More granular support policies make the migration optimization mechanism context-independent.

3.3 Model hypothesis

Content continuity hypothesis: users accessing file exist continuity such as videos, pictures, text messages. When selecting the content of the cache, in addition to calculating the click rate of the file itself, the successively transfer probability p between the extracted files should be calculated. p_{ij} is the probability of extracting j after the file i is extracted.

Node cooperation hypothesis: as shown in figure 2, assuming that the cache capacity of each edge cache node is small, adjacent edge cache nodes cooperate storage. The data transmission time in cooperation nodes is represented as t_{nn}, and t_{nn} is far less than the time of user nodes getting data from the center.

Hierarchical cache hypothesis: when CDN cache replacement is in initiating, in order to ensure getting the prior knowledge of documents hits, the edge caching space of each base station is divided into two parts in proportion: using_part and to_use_part. Since the files with the highest hit rate are stored in using_part, hit rate change is smooth, that part of the documents composition are relatively stable, and file replacement doesn't happen too frequently. On the other hand, to_use_part serves as a secondary cache to store candidates of replacement files, where the replacement of files was accomplished using the classic LRU (Least recently-used) algorithm. From the files stored in this part, through the model calculation, the collection of files with the highest hit rate prediction at the future is selected as the swapping files for using_part.

User mobility hypothesis: there is a cold boot problem when mobile users switch cache nodes. Based on the end-to-end M-FMC model[28], the accuracy of predicting mobility is about 65%. Therefore, it can be assumed that the request and historical request data during the user's movement can basically be transferred to the target node. Using historical data such as the switching and accessing of mobile users can greatly reduce the cold boot problem when users switch to a new cache area.

Model hypothesis: the popularity of the file pop(x_i) is dynamically calculated and broad-

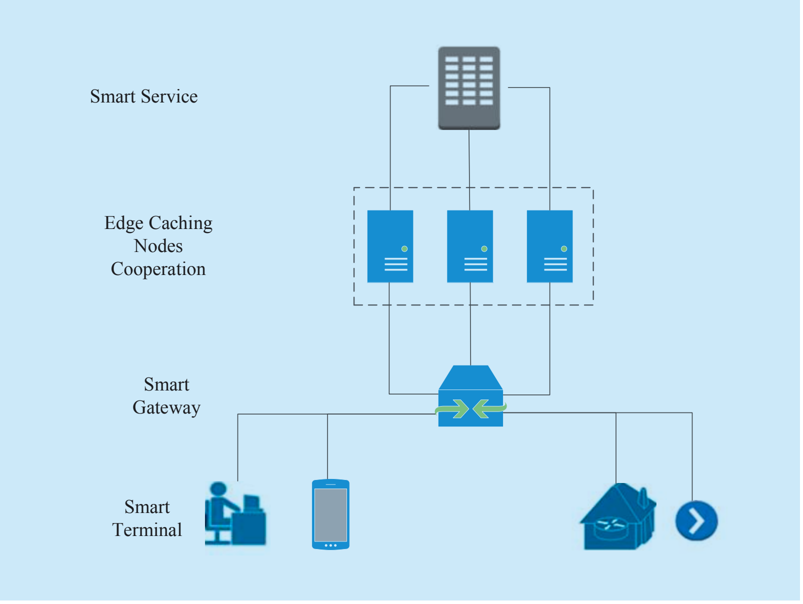


Fig. 2. *Group of edge nodes cooperation.*

casted by the content provider as a known parameter. All resource files are represented as X, X=($x_1 \dots x_n$), c(x) represents the size of space occupied by file x, the cache node space capacity is c, the cache nodes within each coverage area A are denoted as p=($p_1 \dots p_n$), and the transferring time of content from cache node i to terminal j is t_{ij} s, because the cache node is based on base station, and the distance between each base station is similar and short relatively, so we can suppose the t_{ij} directly according to experience.

3.4 Application scenarius

For content providers: when providing users with the data such as videos, pictures or files, in order to improve the user experience and accelerate the speed of data acquisition, content providers cache file of high popularity to the edge cache node. In this way it can achieve the optimal cache effect with limited space and shorter download time with high hit rate.

3.5 Description of prediction model based on markov chain

(1) process discretization: user's access to files is limited and countable; (2) transferring of randomness: user access to web files, with a certain continuity. For example videos, pictures, documents were often browsed in sequence, so visiting process can be represented by a probability value of the history of access log of the system; (3) non-after effect property: the next access state of the user, is related to the current moment, and is considered irrelevant to the earlier state, namely $p(x_n|x_1,\dots,x_{n-1}) = p(x_n|x_{n-1})$.The above property

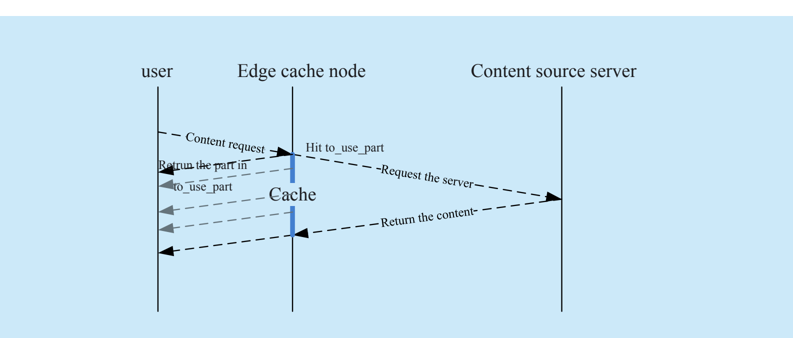


Fig. 3. *To_use_part cache content size.*

conforms to Markov chain model. In every unit time, the extracted joint probability of the collection of files x=($x_1,x_2,\dots,x_n$) is

$$P(x_1,x_2,\dots,x_n) = p(x_1)\prod P(x_n \mid x_1,\dots,x_{n-1}). \tag{1}$$

In order to achieve the best content caching effect of the content provider, which means the content hit rate being the largest and saving the maximum user time. The target function is

$$\max\left(\sum_{i=1}^{n} pop(x_i)\,HitRate(x_i)\right). \tag{2}$$

Among them: pop(x_i)is the file x_i based on the hypothesis of section 3.3, pop(x_i) is known data which is provided by the content provider; HitRate(x) is the weight of file x hit rate, which is defined as:

$$HitRate(x) = \lambda \cdot p(x), \tag{3}$$

where p(x) is the hit rate of file x, and then the weight is indicated by λ. In order to reduce the storage probability of large file and improve the cache hit rate, λ is defined as the open square of the file size

$$\lambda = \sqrt{s(x)} = s(x)^{\frac{1}{2}}. \tag{4}$$

The original target function is:

$$\max\sum_{i=1}^{n} \lambda \cdot s(x_i)^{\frac{1}{2}} \cdot p(x_i). \tag{5}$$

The user extraction file has continuity, which means the next file to be extracted is related to the current file, and is independent of the previously extracted file, which conforms to the Markov chain model, i.e

$$P(x_{s\_n} \mid x_1,\dots,x_{n-1}) = p(x_n \mid x_{n-1}). \tag{6}$$

Take the maximum file extraction probability $p(x_i, x_j)=p(x_i|x_j)p(x_j)$ and the probability $p(x_i|x_j)$ as the key prior knowledge which can be acquired by the statistical probability of the file extraction within unit time, and bring it into the target function to obtain the optimal cache solution.

Obviously, there are more candidate files relative to caching space. In order to store more candidate files in the limited to_use_part space, each candidate file should have a smaller proportion, which means only the beginning of the file is cached. As shown in figure 3, when candidate file request arrives, it will hit the beginning of the cache files firstly, and return this to the user immediately. This part of the data at the same time request source node subsequent resource to save the user's waiting time. The time from cache to server of a request is set as tcs s, and the data transmission speed between edge cache node and content source server is v Mb/s. In theory, the cache size should not exceed 2*tcs*v Mb to realize seamless transmission.

To ensure the reasonableness of file replacement, it is possible to swap files in the using_part section when the expected click rate of files in the secondary cache space is greater than that of the primary cache space. At the same time, the cache space C of the cache node is constant. In order to prevent the cache overflow, it is assumed that the set of files with replacement action is x_{in} and x_{out}.

$$C\left(x_{in}\left(x\right) \leqslant c_{out}\right), x_{in} \subseteq to\_use\_part, x_{out} \subseteq using\_part. \quad (7)$$

When the condition (7) is not satisfied, the cache replacement action is deferred, waiting for x_{in} or x_{out} being updated to satisfy the condition (7).

Based on the above assumptions and formulas, each request data may be updated. The update algorithm is described as follows:

Based on the above algorithm definition, when content is requested, the overall algorithm flow of the edge cache node control module is as follows:

Algorithm 1. Content_replacement.

1. **Input:** using_part, to_use_part, transit_matrix, joint_distribution_matrix, distribution_vec, hit_nums, call_numbs, pop_vec, space
2. **Output:** using_part, to_use_part, next_distribution_vec
3. **The Main parts:**
4. for i in distribution_vec: // update the probability distribution
5.   distribution_vec [i] = distribution_vec[i-1] *0.5 + (hit_numb[i] / call_nums) *0.5 // update weight 0.5, last time result*0.5 + hit probability distribution result*0.5
6. end for
7. for i, j in joint_distribution_matrix: // update the joint probability distribution
8.   joint_distribution_matrix [i, j] = transit_matrix[i, j] * distribution_vec[i]
9. end for
10. for i, j in joint_distribution_matrix:
11.   next_distribution_vec[i] += (joint_distribution_matrix[i, j] + joint_distribution_matrix[j, i])
12. end for
13. min_prob is the min of next_distribution_vec which in using_part label *l*
14. max_prob is the max of next_distribution_vec which in to_use_part label *h*
15. If min_prob < max_prob and space(*h*)<=space(*l*):
    Then update using_part and to_use_part
16. Return using_part, to_use_part, next_distribution_vec

Algorithm 2. The main.

1. **Input**: contents
2. **Initialization**: distribution_vec: [1/n... 1/n].
    transit_matrix: $\begin{cases} \alpha, \alpha > 0.5 \; if \;\; i \to j, \\ \beta, \; \beta < 0.2 \; other\ else \end{cases}$,
    using_part: random continuous files,
    to_use_part: random continuous files,
    hit_nums ={0} for i in contents,
    next_distribution_vec = {0} for i in contents.
3. **Output**: using_part, to_use_part
4. **The Main parts:**
5. While (i = contents. Hasnext)
6.   If i doesn't hit using_part:
7.   Then update to_use_part with LRU
8.   Update using_part, to_use_part, next_distribution_vec with content_swap
9. End while

IV. RESULTS

The purpose of the algorithm in this paper is to improve the hit ratio of cache contents in a certain cache space. Therefore it is assumed that all files are of the same size and with value c1 generally. In hypothesis 2, it is shown that the data transmission time between cooperative nodes time is far less than the time between terminals and source servers. In order to simplify the model, data transmission time between cooperative nodes is ignored because of short distance between cooperative nodes.

Set n_visit as the number of all the accessible files, cache space may be stored to the size of the capacity of *c*. In this algorithm, the ratio of using_part to to_use_part space is 4:1. So the rate of contribution of cache space is only

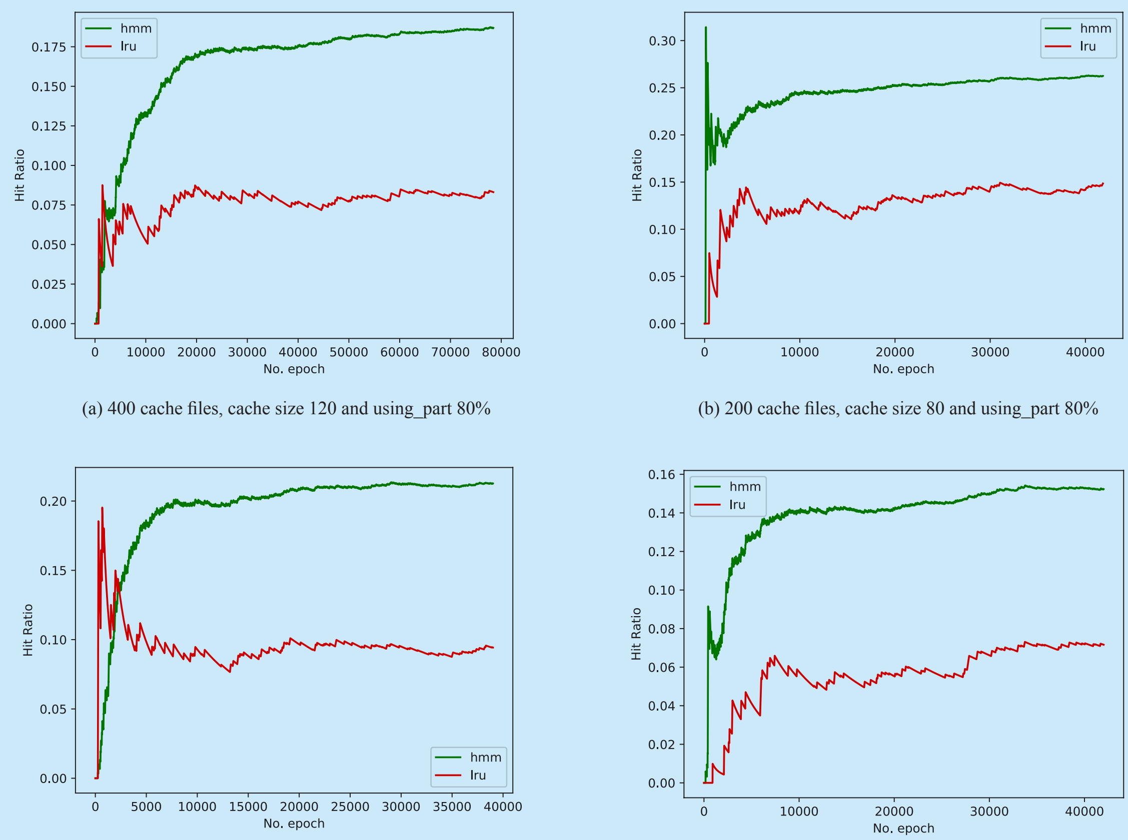
80%, yet LRU algorithm accounts for 100% of buffer space.

Based on the above simplification, to the classic LRU cache replacement strategy, implementation LRU algorithm in this paper under the condition of the same access, with hit rate as a measurement for the experiment. In the case of a 250 files access, hit situation of two algorithms is shown in figure 3.

As can be seen from figure 4, the algorithm caching described in this article is more efficient. Traditional LRU doesn't consider about the correlation between the cache files, so the result of LRU are accidental and has a large fluctuation. However at the beginning of HMM algorithm, there is no historical data, so the key prior knowledge is supposed and the cache file is gotten randomly. As a result, at the beginning in Figure 4 (a) and Figure 4 (c), HMM has a lower cache hit ratio than LRU. In the case of consistent cache space, the cache hit ratio of the algorithm in this paper is almost double as the efficiency of the traditional LRU algorithm. In the control trial, different number of cache files, different cache space sizes, and different using_part proportions were tested. In these three cases, the cache hit ratio of the algorithm described in this paper is nearly 100% higher than that of the traditional algorithm.

V. CONCLUSION

In this paper, the replacement strategy on the edge cache can fuse Markov chain model to improve the hit rate of cache files for blockchain-enabled IoT. At the same time, the edge caching node is able to have strong content



(a) 400 cache files, cache size 120 and using_part 80%

(b) 200 cache files, cache size 80 and using_part 80%

(c) 200 cache files, cache size 60 and using_part 80%

(d) 200 cache files, cache size 60 and using_part 70%

Fig. 4. *The algorithm and LRU hit ratio.*

related to the region according to the geographical features when cache web files. Edge caching nodes in different regions store different contents, which improves the coverage of content caching and thus improves the hit rate. Therefore, algorithm optimization point of this paper is to improve the hit rate of caching contents, thereby optimizing the cache space utilization, and also considering the different parts of the user preferences, reducing huge fluctuations content caused by mobile roaming users. However, there are still some improvements in our work, such as cache content relevance. This algorithm only considers the continuity of the cached content in space, which means the continuity of video files, picture files and text files and etc are not considered the transferring probability caused by the similarity of the file content. For example, the movie files that are not similar in space cause the user's continuous request in the case of the similar subject or the award-winning situation. In addition, besides optimizing cache content utilization, there are also optimization requirements for content providers in terms of cost and energy consumptions, which have not been considered in this paper. Therefore we will study and practice the algorithm from these aspects in the further work.

References

[1] I. Ud Din et al., "The Internet of Things: A review of enabled technologies and future challenges,"*IEEE Access*, vol. 7, 2019, pp. 7606-7640.

[2] Gartner identifies top 10 strategic iot technologies and trends. https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends, 2019.

[3] Sharma P K, Chen M Y, Park J H, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, 2018, pp. 115-124.

[4] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in information-centric networking: Strategies, challenges, and future research directions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, 2018, pp. 1443–1474.

[5] S. Hassan, I. U. Din, A. Habbal, and N. H. Zakaria, "A popularity based caching strategy for the future Internet," *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, 2016, pp. 1–8.

[6] S. Wang, J. Xu, N. Zhang, et al., "A survey on service migration in mobile edge computing," *IEEE Access*, vol. 6, 2018, pp. 23511–23528.

[7] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, 2017, pp. 601–628.

[8] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan et al., "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, 2019, pp. 4660-4670.

[9] X. Xu, X. Zhang, H. Gao et al., "BeCome: Blockchain-Enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, 2020, pp. 4187-4195.

[10] Zhenni Li, Zuyuan Yang and Shengli Xie, "Computing resource trading for Edge-Cloud-assisted Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, 2019, pp. 3661-3669.

[11] Xu, S. Wang, B. K. Bharat, F. Yang, "A blockchain-enabled trustless crowd-intelligence ecosystem on mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol.15, no.6, 2019, pp. 3538-3547.

[12] Wei, W. Feng, C. Zhang, Y. Chen, Y., et al., "Creating efficient blockchains for the Internet of Things by coordinated Satellite-Terrestrial networks," *IEEE Wireless Communications*, vol. 27, no. 3, 2020, pp. 104-110.

[13] M. Xiong, "Research on CDN technology and its application in broadband," *Ph.D thesis, Tianjin university*, 2005.

[14] N. Cai, "Research on the location of the CDN edge node deployment," *Telecommunications Engineering Technology and Standardization*, 2015, pp. 71-76.

[15] X. Wei, S. Wang, Z. Ao, et al., "MVR: An architecture for computation offloading in mobile edge computing," *2017 IEEE International Conference on Edge Computing (EDGE)*, 2017, pp. 232-235.

[16] X. Qiu, L. Liu, W. Chen, Z. Hong and Z. Zheng, "Online deep reinforcement learning for computation offloading in Blockchain-Empowered mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, 2019, pp. 8050-8062.

[17] Xiao, Liang, et al. "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Communications*, vol. 25, no. 3, 2018, pp. 116-122.

[18] Pathan, Mukaddim, R. Buyya, et al., "Content Delivery Networks: State of the art, insights, and imperatives," *Content Delivery Networks*, vol. 9, 2008, pp. 3-32.

[19] D. Boscovic, F. Vakil, S. Dautović, et al., "Pervasive

wireless CDN for greening video streaming to mobile devices," *2011 Proceedings of the 34th International Convention MIPRO,* 2011, pp. 629-636.

[20] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, 2014, pp. 1444-1462.

[21] K. Shanmugam, N. Golrezaei, A. G. Dimakis, et al., "Femtocaching: wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, 2013, pp. 8402-8413.

[22] S. Wang, Y. Zhao, J. Xu, et al., "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, 2018, pp. 160-168.

[23] S. Wang, Y. Zhao, L. Huang, et al., "QoS prediction for service recommendations in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, no. 5, 2017, pp. 134-144.

[24] A. F. Antonescu, A. Gomes, P. Robinson, et al., "SLA-Driven predictive orchestration for distributed cloud-based mobile services," *2013 IEEE International Conference on Communications Workshops (ICC)*, 2013.

[25] T. Taleb, A. Ksentini, "Follow me cloud: interworking federated clouds and distributed mobile networks," *Network IEEE*, vol. 27, no. 5, 2013, pp. 12-19.

[26] T. Taleb, A. Ksentini., "An analytical model for follow me cloud," *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 1291-1296.

[27] A. Ksentini, T. Taleb, M. Chen., "A Markov Decision Process-based service migration procedure for Follow Me Cloud," *2014 IEEE International Conference on Communications (ICC)*,2014, pp. 1350-1354.

[28] A. S. Gomes, B. Sousa, et al., "Edge caching with mobility prediction in virtualized lte mobile networks," *Future Generation Computer Systems*, 2017, pp. 148-162.

Biographies

Hongman Wang, Ph. D., Professor of Beijing University of Posts and Telecommunications, has been long engaged in research and development of network and value-added services. She carries out research work in recent years in the mobile Internet, networking and other aspects of big data, has hosted and participated in more than ten national and provincial projects. She has published more than 40 papers and has more than 20 patents be authorized. Email: wanghm@bupt.edu.cn

Yingxue Li, master student of Beijing University of Posts and Telecommunications. The main research field is service migration and edge caching of mobile edge computing. Email: silviali1225@163.com

Xiaoqi Zhao, graduate student of Beijing University of Posts and Telecommunications. The main research field is network communication and big data applications. Email: xuanzi77@qq.com

Fangchun Yang, Ph. D., Professor of Beijing University of Posts and Telecommunications. Born in Beijing in March 1957. In 1982, 1987 and 1990, he received bachelor, master of computer application and communication and electronic system doctoral degree and taught at Beijing University of Posts and Telecommunications. He served as the director of the trusted network communication Collaborative Innovation Center in 2013. Email: fcyang@bupt.edu.cn