JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 40, 421-433 (2024)
DOI: 10.6688/JISE.202403_40(2).0013

Proactive Caching Strategy Based on Queueing Theory in F-RAN

WEI ZHANG[1,2], HAO-XIANG CHU[1,2] AND HAO HAO[1,2,+]
[1]*Key Laboratory of Computing Power Network and Information Security*
Ministry of Education
Shandong Computer Science Center (National Supercomputer Center in Jinan)
Qilu University of Technology (Shandong Academy of Sciences)
Jinan, 250100 P.R. China
[2]*Shandong Provincial Key Laboratory of Computer Networks*
Shandong Fundamental Research Center for Computer Science
Jinan, 250100 P.R. China
E-mail: wzhang@qlu.edu.cn; 10431210725@stu.qlu.edu.cn; haoh@sdas.org[+]

Fog Radio Access Network (F-RAN) has emerged as a promising architecture to reduce latency and network congestion by caching popular content at the edge. However, optimizing caching strategies in F-RAN faces challenges due to limited storage capacity and global vs. local content popularity. This paper proposes a novel proactive caching placement strategy using queueing theory to minimize latency. Specifically, we formulate an integer linear program based on a queuing model that captures content popularity and service rates. To solve this problem efficiently, we design two low-complexity heuristic algorithms: (1) An improved greedy algorithm that prioritizes globally popular content; and (2) A knapsack algorithm that optimizes cache allocation based on localized content popularity. Extensive simulations demonstrate that our proposed strategy achieves lower average latency and traffic cost compared to baseline caching schemes like LRU, LFU, and random replacement. The key innovation lies in optimizing caching decisions based on joint modeling of queuing delays and localized content popularity. This work provides an effective proactive caching framework for latency-critical F-RAN applications.

Keywords: fog radio access network, the queue theory, proactive caching, content popularity, minimize latency

1. INTRODUCTION

The explosive expansion of mobile networks and IoT devices on edge networks has caused an unprecedented surge in edge data. This increase has intensified the pressure on backhaul links between edge base stations and the core network for content delivery [1-3]. Edge computing has emerged as a promising solution, addressing this issue by caching file content at edge nodes, thereby bringing the content closer to end-users for faster access [4, 5].

Certain applications depend heavily on low latency and high-speed data services. This dependence forces mobile service providers to reconsider their existing network architectures [6] and explore technologies that better accommodate these needs. The Fog Radio Access Network (F-RAN) has been proposed as a potential solution that meets today's Quality of Service (QoS) requirements for low latency, high bandwidth, and high relia-

Received October 31, 2022; revised May 5 & July 15, 2023; accepted August 9, 2023.
Communicated by Xiaohong Jiang.
[+] Corresponding author.

bility in wireless networks [7-9]. Integrating fog computing into existing wireless network architectures allows distributing computing power from cloud computing centers to edge nodes. High-demand content is cached in F-RAN and positioned at fog access points (F-APs) near users. This strategy improves content delivery rates, reduces network traffic costs, ensures the quality of data delivery, and minimizes latency [10]. However, determining the optimal content cache placement policy remains challenging due to the limited storage capacity and communication resources of the edge cache nodes. Previous research has investigated various caching approaches like the Least Frequently Used (LFU) [11], the Least Recently Used (LRU) [12], and the random replacement method. These methods maintain a cache list to track the order or frequency of content access but have limitations when applied to edge computing environments. They must consider the popularity of the content and the network state and resource constraints for targeted caching optimization.

To address these limitations, this paper proposes proactive caching of popular content by predicting the popularity file within the limited caching capacity at the edge nodes. This method has the potential to improve network performance and reduce latency. Given the dramatic increase in data volume and users, applying a low-latency, highly reliable network architecture becomes a necessity. To this end, this paper integrates the F-RAN network framework to exploit its structure for improved caching performance and latency reduction. We focus on active placement policies for cached content in a cache-enabled hierarchical F-RAN network architecture, aiming to minimize latency. We formulate the issue as an Integer Linear Programming (ILP) problem and employ queueing theory for clearer problem definition and treatment. Additionally, two heuristic algorithms are devised for a joint solution to the problem's high computational complexity and content placement process details. The central concept is to understand end-user needs, cache the most suitable content, and make informed decisions.

The remainder of this paper is organized as follows: The theoretical model of the system is presented in Section 2. The design of the system architecture is described in Section 3. The performance evaluation of the model is executed in Section 4, and conclusions are drawn in Section 5.

2. THEORETICAL MODEL

In this section, we present the F-RAN system model equipped with a caching function. The key parameters used are listed in Table 1.

The system model architecture is depicted in Fig. 1. We consider an F-RAN wireless network model that comprises M F-AP nodes, N end-users, and L BBUs. The set of F-AP nodes is denoted by $\mathcal{M} = \{1, \ldots, M\}$, the set of users served by F-AP nodes by $U = \{1, \ldots, N\}$, the set of BBUs by $\mathcal{L} = \{1, \ldots, L\}$, and the requested content of all users by a content library $\mathcal{F} = \{1, \ldots, F\}$. The BBU transmits data through the front-end link to the corresponding F-AP node, each of which is assigned a specific storage capacity, *Cn*. We assume that users can only download the requested content from the F-APs associated with them [13]. Moreover, this study focuses on the preferences of users for the content cached on the F-AP nodes.

Table 1. Key parameters.

Notation	Definition	Notation	Definition
$\mathcal{M}$	Set of F-APs	Y	A binary content cache matrix
M	Number of F-APs	X_{mf}	Content cache decisions
$\mathcal{U}$	Set of mobile users	Y_{lf}	Content cache decisions
N	Number of mobile users	W	The arrival rate
$\mathcal{L}$	Set of BBU	V	The transmission rates
L	Number of BBU	ε_t	Flow through intensity
C_n	Storage budget of F-APs	U	Average download rate
P_f	Content popularity	$\mathcal{T}$	Average time to download content
$\mathcal{F}$	Library of popular contents	D_f	Transfer content files size
F	Total number of contents	R	Content placement decisions
β	The skewness factor of Zipf	H	Traffic cost
P_{mf}	Local content popularity	Z_t	Average time to service a request
X	A binary content cache matrix	O	Traffic cost per link

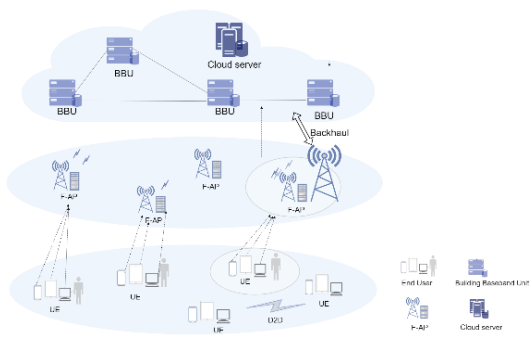

Fig. 1. System model architecture diagram.

3. DESIGN OF THE SYSTEM SCHEME

3.1 Content Popularity

We assume that each end-user requests only a single file of popularity and specify that all file contents D_f are of equal size [13]. If an end-user does not request content, we remove that user from consideration. The Zipf distribution model typically represents the network content popularity preferences of all end-users [14, 15]. We follow this convention in our paper.

$$P_f = \frac{(f)^{-\beta}}{\sum_{j=1}^{F}(j)^{-\beta}}, \forall f \in F \tag{1}$$

where β is the skewness factor. A higher value of β indicates more frequent requests for popular content.

Additionally, different F-APs may exhibit varying file preferences. P_f denotes the probability that the content f is transmitted on F-AP node f. In this context, $P_f = \sum_{m=1}^{M} P_{mf}$.

3.2 Basic Settings

In this paper, we introduce queueing theory and design a shared queueing model to represent the arrival rate. Fig. 2 illustrates the three service request models F-AP way, BBU way, and Fronthaul way, controlled by a common task queue. The first path depicts the end-user sending a request for content to the corresponding F-AP node. If not found, a lookup is performed in the BBU. If still not found, the content is cached in the cloud center server and passed back through the third pattern shown in the figure. W_m, W_l, and W_n represent the arrival rates of the three communication modes, respectively, which we will discuss further.

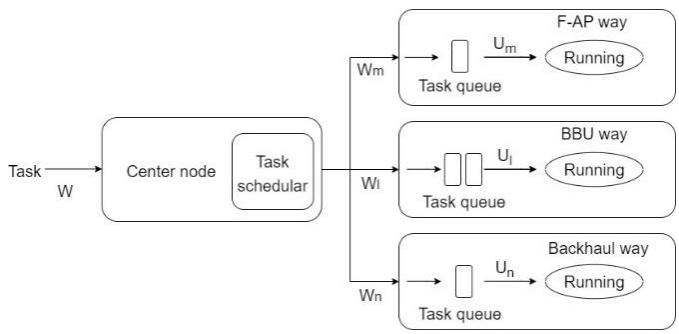

Fig. 2. Theoretical model.

This section treats the problem as a content placement model aiming for minimal latency. The storage allocated to the cache of F-AP n is denoted as Cn. We define two Boolean variables, X and Y, and two binary content cache matrices. $X = \{X_{mf} | m \in M, f \in F\}$ signifies whether content f is stored in the local cache of the fog server. If content f is precached in fog server n, we assign $X_{mf} = 1$, otherwise $X_{mf} = 0$. Similarly, for Y_{lf}, denoted as $Y = \{Y_{lf} | l \in M, f \in F\}$, it indicates whether the BBU caches the f_{th} file. If the BBU caches the f_{th} file, we assign $Y_{mf} = 1$, otherwise, we assign $Y_{lf} = 0$.

3.3 Problem Formulation

This paper primarily considers wireless transmission delay and backhaul delay components. We model this scenario as a queueing system with a Poisson arrival rate and an exponential service rate (M/M/1 queueing process) representing each transmission mode. Furthermore, we use W to denote the shared queueing of arrival rates [13]. In this paper, W_m, W_l, and W_n denote the three communication modes, respectively. O_m, O_l, and O_n represent the traffic cost in each transmission process, and V_m, V_l, and V_n represent the service rates. Eq. (2) presents the arrival rate through the associated F-AP node m.

$$W_m = W \sum_{f=1}^{F} P_f X_{mf} O_m \tag{2}$$

where $\sum_{f=1}^{F} P_f(1 - X_{mf})$ denotes the probability that content f is requested and cached in the associated F-AP m, corresponding to a service rate of $V_l = \frac{1}{T_{mf}^L}$.

$$W_l = W\sum_{f=1}^{F} P_f(1-X_{mf})Y_{lf}O_l \tag{3}$$

Eq. (3) represents the arrival rate of the content cached through the BBU. $\sum_{f=1}^{F}P_f(1-X_{mf})$ represents the probability that content f will be cached in the corresponding BBU and not stored in the associated F-AP node m. The corresponding service rate is $V_m = \frac{1}{T_{mf}^M}$.

$$W_n = W\sum_{f=1}^{F} P_f(1-X_{mf})(1-Y_{lf})O_n \tag{4}$$

Finally, Eq. (4) presents the arrival rate via the backhaul. $\sum_{f=1}^{F}P_f(1-X_{mf})\ (1-Y_{lf})$ indicates that content f can only be found or stored in the backhaul and is inaccessible via other service modes. The corresponding service rate is $V_n = \frac{1}{T_{mf}^N}$.

In queueing theory, we use the ratio of the average service time to determine the traffic intensity during transmission and the average delivery time to represent its arrival interval. This paper defines this ratio as the traffic intensity of a t-level cluster, expressed in Eq. (5). Its value is related to the three communication modes' arrival rates W_m, W_l, W_n, and service rates V_m, V_l, and V_n. To stabilize the global delay condition, we set the value of ε to be less than 1 [13].

$$\varepsilon_t = \frac{W_m}{V_m} + \frac{W_l}{V_l} + \frac{W_n}{V_n} \tag{5}$$

The goal is to minimize the average end-to-end latency. We assume that end-users with the same request belong to the same cluster and that the entire content directory is cached in the core network [16-18]. A user sends a request to the corresponding F-AP node and performs a lookup with an average download rate of U_m^M. If the request is unsuccessful, the lookup is performed in the adjacent F-AP node [19]. Otherwise, the request is sent to the BBU for lookup and transmitted at $U_{m,l}^L$. If the fetch is still unsuccessful, the content is fetched from the core network and disseminated through the BBU, with an average download rate of $U_{m,n}^N$ and the file size of the transmitted content as D_f.

Subsequently, we use the following equations to clarify the transmission time for various service modes:

$T_{mf}^M = \frac{D_f}{U_m^M}$ represents the average time for end-users associated with the F-AP nodes to download the transmitted content f from the F-AP nodes.

$T_{mf}^L = \frac{D_f}{U_m^M} + \frac{D_f}{U_{ml}^l}$ represents the average time for the end-users associated with the F-AP nodes to download the transmitted content f from the corresponding BBU.

$T_{mf}^N = \frac{D_f}{U_m^M} + \frac{D_f}{U_{m,l}^l} + \frac{D_f}{U_{m,N}^N}$ represents the average time for the end-users associated with the F-AP nodes to download the transmitted content f from the core network center.

Considering delay and interference during transmission, this paper assumes that the average download rate size relationship for each service mode is $U_{m,n}^N > U_{m,l}^L > U_m^M$.

We further define the average time Z_t for each service request, as shown in Eq. (6), which is determined by the arrival and service rates. The second half is the average size of the queueing system W^{i-1} for the arrival rate we set. Here, ε_t represents the traffic intensity, and for each set i, W_i represents the arrival rate, and V_i represents the service rate.

$$Z_t = \frac{\varepsilon_t}{W_t} + \frac{\varepsilon_t = \frac{W_m}{V_m^2} + \frac{W_l}{V_l^2} + \frac{W_n}{V_n^2}}{1 - \varepsilon_t} \tag{6}$$

In summary, the weighted average network delay for each service request can be expressed as shown in Eq. (7). Constraint C_1 indicates that the maximum value of the storage budget in each node is C. Constraints C_2 and C_3 represent the caching decisions of the fog server.

$$\begin{aligned} &\min_{X_{mf} Y_{lf}} Z = \frac{1}{W} \sum_{t=1}^{T} W_t Z_t \\ &\text{s.t.} C_1 : \sum_{n=1}^{N} c_n \le C \\ &C_2 : X_{mf} \in \{0,1\}, \forall m \in N, \forall f \in F \\ &C_3 : Y_{lf} \in \{0,1\}, \forall l \in N, \forall f \in F \end{aligned} \tag{7}$$

We generalize this to an integer linear programming (ILP) [20] problem. However, due to high computational complexity, the system performance is poor. Therefore, we design two low-complexity suboptimal algorithms in the following sections to improve the overall performance.

3.4 Problem Solution

The high computational complexity of the target formulation necessitates the design of two suboptimal heuristics to address the issue while effectively managing time complexity and enhancing efficiency. Initially, the content popularity plays a crucial role in formulating the caching policy for each cache entity, where higher popularity translates to better content performance. We have accordingly improved the greedy algorithm to maximize the cache of popular content at each cache node. This is guided by global popularity, with the following detailed procedure: We deploy a greedy algorithm, enabling each F-AP node to cache the most popular content until the cache storage capacity is reached. Algorithm 1 presents a detailed description of the process.

Subsequently, a knapsack algorithm is proposed to take into account the local content popularity of each F-AP node, aiming to mitigate network traffic and reduce latency.

Algorithm 1: Improved heuristic algorithm

Input: M, F, N, P_{mf}, C_n, O_m, O_l, O_n, D_f;
Output: Content placement decisions R, Traffic cost O;

1: **for** F-AP m **do**
2: Storage content available to end users D_f;
3: **for** order of popularity of content P_f **do**
4: Make the most popular content cached on each F-AP node and the storage capacity full to make stop.
5: **end for**
6: To get content placement decision $X_{m,f}$;

7: **end for**
8: $X_{m,f}$ can be expressed as element R;
9: Substitute M, F, N, P_{mf}, C_n, O_m, O_l, O_n, D_f, R into Eq. (7) to calculate O;
10: **return** R, O;

The fog server, based on content popularity P_{mf} and content size D_f, executes caching decisions. The decision process can be represented as follows,

$$\max_x \sum_{f=1}^{F} P_{mf} D_f X_{mf}. \tag{8}$$

Furthermore, the process must satisfy $\sum_{f=1}^{F} x_{mf} D_f - c_n \leq 0$, signifying that the total content cached in each F-AP must not exceed its capacity limit. As demonstrated by Eq. (8), content of higher popularity is preferred for caching in the fog server. We model this as a 0-1 knapsack problem [21], where C_n represents the given knapsack capacity, D_f indicates the weight of the content items, and X_{mf} is a content placement decision variable.

Solving this problem necessitates finding a recursive relationship between the original problem and the subproblem. Thus, we construct a matrix B. With the content items to be cached being denoted as j (where j is in {1, …, f}), we can use $B(f, j)$ to represent the maximum objective value of the obtained content. The optimal solution, therefore, is $B(f, C_n)$. The corresponding system of relational equations is as follows,

$$B(f, j) = \begin{cases} B(f-1, j), & \text{if } j < D_f \\ \max\{B(f-1, j), B(f-1, j-D_f) + P_{mf} D_f\}, & \text{otherwise} \end{cases}. \tag{8}$$

As specified in Eq. (9), when the storage capacity size j is less than the content item D_f, the fog server doesn't cache the content. Following previous work [22], we exclude the content f from this scheme, considering only {1, …, $f-1$} in the cached data, represented as $B(f-1, j)$. Alternatively, the decision lies between caching content f or excluding it from the cached data. The first term in the max function corresponds to excluding f from the cache, while the second term denotes including the content f in the F-AP node cache. In the latter case, the value $P_{mf} D_f$ is added to the total value, and D_f storage space is occupied.

Algorithm 2: Local Content Popularity Knapsack Part Algorithm

Input: M, F, N, P_{mf}, C_n, O_m, O_l, O_n, D_f;
Output: Content placement decisions R, Traffic cost O;
1: **for** F-AP m **do**
2: **for** $f = 1 \rightarrow$ **F do**
3: **for** $i = 1 \rightarrow C_n$ **do**
4: **if** $D_f > i$ **then**
$B(f+1, i+1) = B(f, i+1)$;
5: **else**
6: $B(f+1, i+1) = \max\{B(f, i+1), B(f, i+1-B_f) + P_f D_f\}$;
7: **end if**
8: **end for**
9: **end for**

10: $b = B(F + 1, C_n + 1)$, $f = F$, $i = C_n$;
11: **end for**
12: X_{mf} is the element of R; Substitute M, F, N, P_{mf}, C_n, O_m into Eq. (7) to calculate O;
13: **return** R, O;

Algorithm 2 provides a comprehensive illustration of the knapsack algorithm process, rooted in local content popularity. We can compute all elements in B and then backtrack these elements in B to determine the content cached in the F-AP nodes, based on the optimal solution.

4. PERFORMANCE EVALUATION

4.1 Simulation Environment

In our simulation, we configure the number of F-AP nodes ($\mathscr{M}$) to be 20, and the number of end-users (u) to be 800.The quantity of transmitted content, F, is set to 100. The size of each transmitted content, D_f, is configured to 10 MB, and the capacity of the AP nodes is set to 1500 MB. We set the traffic cost to the fog server, BBU pool, and core network as O_m, O_l, and O_n, respectively, with $O_m < O_l < O_n$. As per the parameter configurations cited in literature [22], O_m is configured as 1 MB, O_l is set to 2 MB, and the O_n value is set to 4MB. The average size of the transmitted content is D_f, and the actual content size in the network varies randomly between 0.5 D_f and 1.2 D_f . The popularity of content in the transmission network is described by a Zipf distribution with $\sigma = 0.56$. We denote the probability of a user requesting content f at F-AP nodes by P_{mf}. The parameter settings are demonstrated in Table 2.

Table 2. Key parameters.

Parameter	Value	Parameter	Value
Number of users	$u = 800$	The average content size	D_f =10 MB
Number of F-AP	$\mathscr{M} = 20$	The arrival rate of W Zipf	1MB/s
The traffic cost of O_m, O_l, O_n	1,2,4MB	distribution skewness	$\sigma = 0.56$

4.2 Dataset

To validate the performance of this cache placement in a real-world environment, we employ the MovieLens 1M and MovieLens 100K datasets [23]. These datasets contain authentic ratings from numerous users across various movies. We simulate the hypothetical user-requested content process by treating movies with user participation ratings as user-requested content and each movie rating as a unique content download. Similar content request simulation approaches have been used in literature [22, 24].

4.3 Evaluation and Discussion

To study the impact of cache size on delivery latency, we assess the latency of different testing methods at various average content sizes. This evaluation involves a comparison of the proposed scheme (Que-AL) with five others, namely Thompson sampling, Random schemes, Queue, and NoStrgAlloc. Fig. 3 illustrates the relationship between average content delivery latency and cache size.

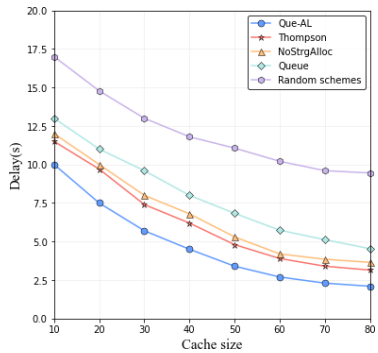

Fig. 3. Delay vs Cache size.

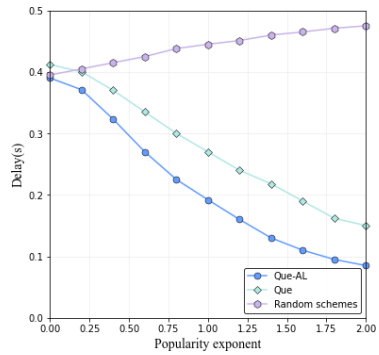

Fig. 4. Delay vs Popularity exponent.

For the Random schemes, which involve random content selection for caching, performance suffers due to disregard for memory allocation and content popularity. Insights from NoStrgAlloc and Queue underscore the equal importance of considering the storage resource allocation of the fog computing server, based on content following the Zipf distribution. Early sampling using probabilistic models, as in the Thompson Sampling algorithm, carries some advantages, but these are relatively limited. The algorithm proposed in this paper amalgamates local content popularity distribution with a sound, sample-based cache design to achieve optimal results.

We further evaluate the impact of the content popularity index on the system and compare our proposed scheme with Queue and Random schemes. Fig. 4 illustrates the relationship between content delivery delay and the popularity index. The performance of our proposed scheme improves as the popularity index of the requested file increases. This scheme shows symmetric variation as the popularity index continues to grow, which signifies that only a small portion of the content is in high demand. The best latency is achieved when user-requested content is stored in the cache as much as possible. This explains the improved results of our method as the popularity index increases. The Queue scheme does not adapt well to changes in content since it fails to distinguish between different contents. On the other hand, the Random scheme has a low probability of finding requested content in the local cache due to its random content caching strategy.

To further assess the performance of the proposed algorithm with different average transmission content sizes, we conducted experiments with the total storage capacity (C) of all F-AP nodes in the transmission network set to 1500 MB.

Figs. 5 and 6 show the traffic cost incurred when the average content size is transferred from 5MB to 12MB; it can be seen that as the average content size increases, the traffic cost also increases. This is because the larger the size of the transmitted content, the more network traffic is generated. The Random schemes has the highest traffic overhead due to the high randomness, and the standard Queue has the second highest traffic overhead. The Thompson sampling algorithm gives better results due to sampling. In addition, the algorithm proposed in this paper starts from local popularity allocation, which avoids the disadvantage of considering only global popularity. The comparison between Fig. 5 and Fig. 6 shows that the experimental results are better when the data set is 1M due to the enlightenment of the algorithm in this paper.

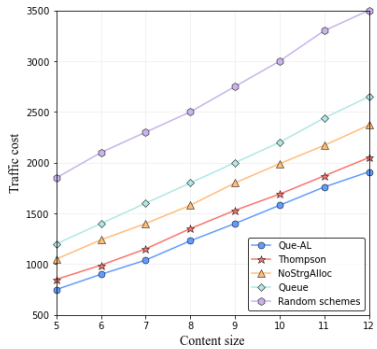

Fig. 5. Traffic cost vs content size (1M).

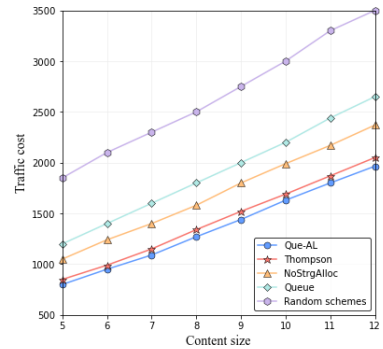

Fig. 6. Traffic cost vs content size (100K).

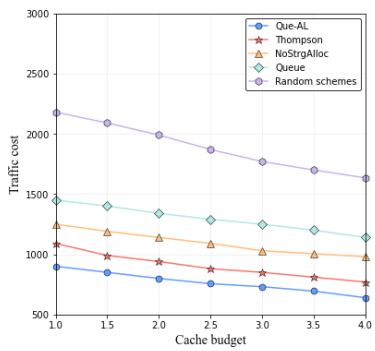

Fig. 7. Traffic cost vs Cache budget (1M).

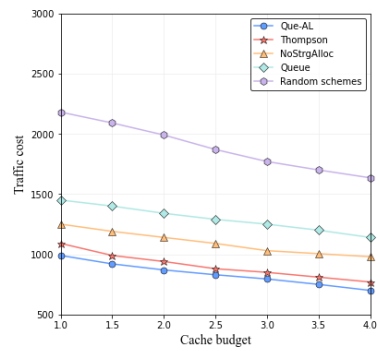

Fig. 8. Traffic cost vs Cache budget (100K).

Since the storage budget of the cache is an important metric when designing a caching strategy. Figs. 7 and 8 show the relationship between the traffic cost and the F-AP cache budget we tested on the MovieLens 1M and MovieLens 100K datasets, respectively. From the figures, we can observe that the traffic cost decreases as the caching budget of the F-AP nodes increases because more popular content can be cached in the F-AP nodes as the caching budget increases. Therefore, the traffic on the other transport links decreases in this case. First, considering the importance of local content hotness for caching files, the performance of Queue and Random schemes could be better. The better performance of this method compared to the Thompson Sampling and NoStrgAlloc is because the other methods consider the local content hotness rather than the global popularity. By comparing Figs. 7 and 8, we can see that if the training involves more datasets, better results can be achieved, and the performance of the proposed algorithm is closer to the optimal algorithm.

5. CONCLUSIONS

This paper introduces a dynamic content caching scheme that integrates queuing theory to address the issue of latency-constrained content caching in edge environments. In the F-RAN network architecture scenario, the transmission process experiences high network traffic and significant latency throughout the communication process due to cache

capacity limitations. Consequently, we aim to control latency to reduce the total network traffic cost of transmitting user-requested content. Through analysis, we frame this issue as an integer linear programming problem and apply queuing theory to more accurately formulate and address the problem. We design two heuristic algorithms to jointly solve the problem, considering its high computational complexity and the specifics of the content placement process. Experimental results demonstrate that the proposed method outperforms previous caching algorithms in this context. There are still many points that could be enhanced in the experimental design strategy, such as considering the capability of the aggregation nodes in the cloud and considering more metrics factors. Future work will focus on the application and implementation of new edge scenarios.

ACKNOWLEDGMENT

This work was supported in part by the Project of Key R&D Program of Shandong Province No.2022CXGC20106, the Shandong Provincial Natural Science Foundation under Grant No.ZR2022QF040 and No. ZR2020LZH010, the Pilot International Cooperation Project for Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant No. 2022GH007 and No. 2022JBZ01-01, the QLU Pilot Project of Integration of Science, Education and Production under grant 2022PX083, the National Natural Science Foundation of Shandong Province under grant ZR2022QF040 and the QLU Pilot Project of Integration of Science, the One Belt One Road Innovative Talent Exchange with Foreign Experts under Grant No. DL2022024004L, and the Jinan Scientific Research Leader Studio Project under Grant No. 2021GXRC091.

REFERENCES

1. Z. Kuang, Y. Shi, S. Guo, J. Dan, and B. Xiao, "Multi-user offloading game strategy in ofdma mobile cloud computing system," *IEEE Transactions on Vehicular Technology*, 2019, pp. 12 190-12 201.
2. Y. Wang, Y. Dong, S. Guo, Y. Yang, and X. Liao, "Latency-aware adaptive video summarization for mobile edge clouds," *IEEE Transactions on Multimedia*, 2020, pp. 1193-1207.
3. C. Xu, "Context aware mobility in internet of things enabling technologies, applications, and challenges," *Transactions on Emerging Telecommunications Technologies*, 2022, p. e4624.
4. X. Xia, F. Chen, Q. He, J. C. Grundy, M. Abdelrazek, and H. Jin, "Cost-effective app data distribution in edge computing," *IEEE Transactions on Parallel and Distributed Systems* 2021, pp. 31-44.
5. Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, "Federated learning based proactive content caching in edge computing," in *Proceedings of IEEE Global Communications Conference*, 2018, pp. 1-6.
6. M. Bhatia, "Energy efficient IoT-based informative analysis for edge computing environment," *Transactions on Emerging Telecommunications Technologies*, Vol. 33, 2022, p. e4527.

7. J. Wu, C. Yang, and B. Chen, "Proactive caching and bandwidth allocation in heterogenous networks by learning from historical numbers of requests," *IEEE Transactions on Communications,* 2020, pp. 4394-4410.
8. M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Network,* 2016, pp. 46-53.
9. R. Verma and S. Chandra, "Fogbus3: A scalable and reliable framework for integrated IoT and fog computing scenario," *Transactions on Emerging Telecommunications Technologies,* Vol. 33, 2022, p. e4535.
10. M. Shirmohamadi, H. Bakhshi, and M. Dosaranian-Moghadam, "Optimizing resources allocation in a heterogeneous cloud radio access network using machine learning," *Transactions on Emerging Telecommunications Technologies,* Vol. 33, 2022, p. e4570.
11. M. M. Uddin and J. Park, "360 degree video caching with LRU & LFU," in *Proceedings of IEEE 12th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference,* 2021, pp. 45-50.
12. W. Xiong, S. Katzenbeisser, and J. Szefer, "Leaking information through cache LRU states in commercial processors and secure caches," *IEEE Transactions on Computers,* 2021, pp. 511-523.
13. S. Mrad and S. Hamouda, "Proactive caching placement strategy with end-to-end delay reduction in c-ran using queuing theory," in *Proceedings of the 5th International Conference on Advanced Systems and Emergent Technologies,* 2022, pp. 91-96.
14. X. Fan, H. Zheng, R. Jiang, and J. Zhang, "Optimal design of hierarchical cloud fog & edge computing networks with caching," *Sensors,* Vol. 20, 2020, p. e1582.
15. P. Blasco and D. Guenduez, "Learning-based optimization of cache content in a small cell base station," in *Proceedings of IEEE International Conference on Communications,* 2014, pp. 1897-1903.
16. J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Cache placement in fog-rans: From centralized to distributed algorithms," *IEEE Transactions on Wireless Communications,* 2017, pp. 7039-7051.
17. M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Transactions on Wireless Communications,* 2016, pp. 6118-6131.
18. Y. Fukushima, T. Suda, T. Murase, Y. Tarutani, and T. Yokohira, "Minimizing the monetary penalty and energy cost of server migration service," *Transactions on Emerging Telecommunications Technologies,* Vol. 33, 2022, p. e4511.
19. T. Xiao, T. Cui, S. M. R. Islam, and Q. Chen, "Joint content placement and storage allocation based on federated learning in F-RANs," *Sensors,* Vol. 21, 2021, p. e215.
20. F. Aloul, A. Ramani, I. Markov, and K. Sakallah, "Generic ILP versus specialized 0-1 ilp: An update," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design – Digest of Technical Papers,* 2002, pp. 450-457.
21. A. Freville, "The multidimensional 0-1 knapsack problem: An overview," *European Journal of Operational Research,* Vol. 155, 2004, pp. 1-21.
22. S.-H. Park, O. Simeone, and S. Shamai (Shitz), "Joint optimization of cloud and edge processing for fog radio access networks," in *Proceedings of IEEE International Symposium on Information Theory,* 2016, pp. 315-319.
23. F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM*

Transactions on Interactive Intelligent Systems, Vol. 5, 2015, pp. 1-19.
24. S. Mueller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Transactions on Wireless Communications*, Vol. 16, 2017, pp. 1024-1036.

Wei Zhang received the B.E. degree from Zhejiang University in 2004, the MS degree from Liaoning University in 2008, and the Ph.D. degree from Shandong University of Science and Technology in 2018. He is currently a Professor with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences). His research interests include future generation network architectures, edge computing and edge intelligence.

Hao-Xiang Chu is a master's student with the Shandong Computer Science Center (National Supercomputing Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences). His research interests include edge computing, computing network convergence and future generation network architectures.

Hao Hao received the Ph.D. degree in Computer Science and Technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2021. He is currently a Lecturer with the Shandong Computer Science Center (National Supercomputing Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences). His research interests include MEC and content caching over the wireless network, multimedia communications.