Received December 9, 2019, accepted December 26, 2019, date of publication January 6, 2020, date of current version January 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964258

Dueling Deep-Q-Network Based Delay-Aware Cache Update Policy for Mobile Users in Fog Radio Access Networks

BOREN GUO, XIN ZHANG, QIWEI SHENG, AND HONGWEN YANG

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Xin Zhang (zhangxin@bupt.edu.cn)

The research was supported by National Science and Technology Major Project with grant No. 2018ZX03001024-006.

ABSTRACT Fog radio access networks (F-RANs) can effectively alleviate fronthaul loads and reduce content transmission delay by migrating cloud services to the network edge. This paper addresses a cooperative caching scenario in F-RAN, where each mobile user can acquire the requested contents from any one of its associated fog-computing-based access points (F-APs). However, caching disparate contents in different F-APs will lead to different content delivery delays, since mobile users suffer from diverse channel fadings and interferences when they download contents from different F-APs. Considering limited caching storage in each F-AP, diverse user preferences, unpredictable user mobility and time-varying channel states, an average transmission delay minimization problem is formulated. With the aid of dueling deep-Q-network framework, a delay-aware cache update policy is proposed for mobile users in F-RAN. The proposed cache update policy will decide to replace the stored contents in F-APs with the proper contents at each time slot. Compared with first in first out, least recently used and least frequently used caching policies, simulation experiments are performed to evaluate the performance of the proposed algorithm. Simulation results illustrate that the proposed caching policy yields better average hit ratio and lower average transmission delay than other traditional caching policies.

INDEX TERMS Caching, fog radio access network, hit ratio, mobility, reinforcement learning.

I. INTRODUCTION

Driven by the rapid advance of diverse smart devices and various multimedia applications, the mobile data traffic over wireless network has experienced a tremendous growth. In the Cisco white paper [1], the global mobile data and Internet traffic is predicted to grow at compounded annual growth rate of 46%, which will impose many serious issues on wireless network, e.g., network congestion, server overload and so forth. Although uncountable multimedia data surges from different services, e.g, Internet of things [2], network slicing [3], wireless-powered communication [4], device to device (D2D) communications [5], etc., there are numerous redundant and repeated contents. Caching the popular contents in the centralized baseband unit (BBU) pool of cloud radio access network is an effective approach to reduce redundant and repeated data, but capacity-limited fronthaul links still suffer from a large number of content requests from various applications. To relieve the pressure on fronthaul links, fog radio access network (F-RAN) as a promising architecture has been proposed [6]. The access points in F-RAN, also named fog-computing-based access points (F-APs), are equipped with fog-computing units, storage resources and part of baseband processing functions, so as to cache the most popular contents at the network edge. By storing the contents closer to the requesting mobile users (MU), the fronthaul load can be alleviated effectively. However, how to make full use of the computation resources and storage capacities in the F-APs has attracted more and more attentions from researchers. In addition, owing to the time-varying user characteristics including content preferences and user mobility, what, when and where to cache has been one of the hottest issues in recent years.

Generally, content caching includes cache placement [7]–[14] and cache update [15]–[17]. Specifically, the cache placement policy figures out what should be stored, whilst the cache update policy addresses when to store. To solve the cache placement problem, researchers devote to predict the content popularity. Then, the most popular contents are placed in the local cache, and the stored contents

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

are unchanged for a long time. In the cache update policy, a requested content should be stored at a proper time slot. Therefore, the requested content can be delivered in time, when the request occurs at next slot. The stored contents may be different at each slot. Considering the time-varying user preferences, cache update is a feasible way to maximize the long-term average hit ratio.

In addition, cooperative caching [11], [18]–[22] is an effective way to improve the cache space utilization. The cooperative caching means that the requested contents can be obtained from multiple content providers via content sharing and other manners. Since each MU in the F-RAN system can be served by multiple F-APs, it can get the contents from any one of its associated F-APs. Storing the requested contents in different F-APs will leads to different cache hit ratios. Therefore, where to store the requested contents is also a significant problem for the researchers concerned about the cooperative caching.

Although the cooperative caching can effectively enhance the cache space utilization, the unpredictable mobility of MUs has a significant impact on the utilization. Since the topological relation between the MUs and their associated F-APs is time-varying, some of the stored contents may not be requested by the new incoming MUs. Consequently, the stored contents should be updated timely to meet the demands of MUs.

Besides, the content delivery can also heavily affect the caching policies, especially for the delay-sensitive services. In F-RAN, when an MU downloads contents from its associated F-APs, the MU may suffer from different channel fadings and interferences, which will result in different content transmission delays. To achieve the minimum average transmission delay, the caching policy should decide which F-APs the requested contents should be stored in.

This paper considers a cooperative content caching and delivery scenario for MUs in the F-RAN system. In such case, user preferences and channel states are time-varying, the mobility pattern of each MU is unpredictable. In order to minimize the average transmission delay of the requested contents, how to store contents in the F-APs is a complicated problem. Inspired by the success of machine learning applying in various fields [23], [24], a deep reinforcement learning (DRL) framework, dueling deep-Q-network (DQN) [25] is employed to settle the problem above. Notably, a dueling DQN based delay-aware cache update policy is proposed. Compared with three traditional caching policies, i.e., first in first out (FIFO), least recently used (LRU) and least frequently used (LFU), the performance of the proposed caching policy is evaluated through simulations and analyses. The main contributions can be drawn as follows:

- Taking into account time-varying user preferences, unpredictable user mobility, cooperative caching between adjacent F-APs and different channel states, including channel fading and interference, an average transmission delay minimization problem is formulated.
- To address the optimization problem above, the cache update is modeled as an Markov decision process (MDP). Then, dueling DQN technique is adopted to deal with the MDP problem without any priori knowledge of state transition probability. Finally, a dueling DQN based delay-aware cache update policy is proposed.
- In comparison with FIFO, LRU and LFU caching policies, the performance of the proposed caching policy is validated in terms of average hit ratio and average transmission delay.

The rest of this paper is organized as follows. The related works are discussed in the next section. Section III presents the system model of cooperative content caching and delivery in F-RAN. In Section IV, a dueling DQN based delay-aware cache update policy is proposed. Finally, Section V concludes this paper.

II. RELATED WORKS

The content caching problem, including cache placement and cache update, has attracted researchers from many fields, e.g., D2D communications [13], [16], [21], [22], [26], [27], F-RAN [12], [28], [29], mobile edge computing [7], [18], [30] and so on.

As for the cache placement [7]–[14], researchers focus on how to obtain the content popularity and user characteristics, e.g., content preference, quality of experience (QoE), mobility and so forth, so as to proactively cache the most popular contents. Authors in [7] proposed three hierarchical edge caching mechanisms, including random caching, proactive caching and game-theory-based caching, for 5G edge computing mobile multimedia wireless networks, where popular multimedia contents can be cached at routers, base stations or mobile devices. Considering a tradeoff between cache hit ratio and occupied cache space, research in [8] studied the cache space efficient caching in content-centric mobile ad hoc networks. Considering the different rate-distortion characteristics of videos and the coordination of cache providers, [9] addressed a mobile edge cache placement optimization problem via greedy algorithm. Taking into account users' diverse demands over different locations, [10] proposed location customized caching schemes. Besides, two popularity prediction algorithms are developed for two noise models. By using deep learning, authors in [11] proposed two proactive cooperative caching algorithms to predict user preferences in a centralized way and a distributed way, respectively. By learning user preference, two edge caching architectures are proposed to predict content popularity in [12]. By applying transfer learning technique, the knowledge of user preference and activity level can be learned to optimize the caching policy in D2D communications [13]. With the aid of the rating matrix, Cheng et al. proposed a Bayesian learning method to estimate the individual content request probability, which reflects personal preferences. Then, the estimated request probability is incorporated into caching strategy to optimize system throughput [14].

For the cache update [15]–[17], researchers look for policies to maximize the long-term average hit ratio. For the first time, Zhong et al. employed DRL framework to make content replacement decisions to maximize the hit ratio [15] for a single base station. Employing multi-agent RL technique, Jiang et al. proposed a content caching strategy in D2D networks [16]. Considering the space-time popularity of requests and cache-refreshing costs, authors in [17] proposed a Q-learning caching algorithm for 5G cellular networks.

Besides, cooperative caching [11], [18]–[22] is an effective approach to improve the utilization of storage resource. Researchers in [18] focused on a cooperative edge caching architecture for content-centric 5G networks, and proposed a mobility-aware caching framework for MUs. Lin et al. focused on cooperative caching in the heterogeneous ultradense network, which includes coordinated multipoint-integrated ultradense cells and cluster-based device-to-device (D2D) networks [19]. In [20], Zhou et al. proposed a cooperative probabilistic caching strategy in a spatially clustered cellular networks scenario, where base stations within a cluster can share cached contents with each other. Wu et al. studied which content should be cached and which requester is important, and proposed a distributed collaborative cache management scheme for D2D communications [21]. Taking into account users' similarity in accessing videos, the work in [22] built a cooperative cache list to determine what videos need to be cached.

The works in [26], [28], [29] and [31] not only focus on the content caching, but also consider the content delivery. Authors in [26] designed a non-parametric estimator to learn the intensity function of requests, and then proposed a learning-based caching algorithm in D2D-enabled networks. [28] presented a mobile virtual reality delivery framework in the fog radio access network, and a joint caching and computing policy is proposed to optimize resource allocation. Li et al. constructed a fog-community architecture for content caching in D2D enabled F-RAN from the social view point [29]. A theoretical framework is proposed in [31] to characterize the tradeoff among computing, cache and communication resources for content delivery in the mobile edge network.

Moreover, some researchers [32]–[34] address economical efficiency and energy-efficient caching policies. To provide different services for users with different requirements, the authors in [32] investigated the optimal economical caching schemes in cache-enabled heterogeneous networks. To minimize the energy consumption of the network, authors in [33] employed an integer linear programming optimization model to evaluate energy benefits and proposed a heuristic algorithm to power-on and power-off caches. Taking into account the energy cost of downloaded contents and channel quality, Somuyiwa et al. proposed a threshold-based proactive caching scheme to minimize the long-term average energy cost [34].

In addition, considering the mobility of vehicles, researchers in [35], [36] tried to predict the movement of vehicles, so that the contents can be stored in the next associated road side unit in advance. Zhang et al. investigated the caching problem of multi-view 3D videos in the 5G networks [36], and an actor-critic, model-free algorithm is adopted to find the effective proactive caching policy. To improve the QoS for non-safety related services, a Q-learning-based proactive caching strategy for vehicular networks is proposed in [35].To the best knowledge of the authors, few studies have considered both unpredictable user mobility and time-varying channel states.

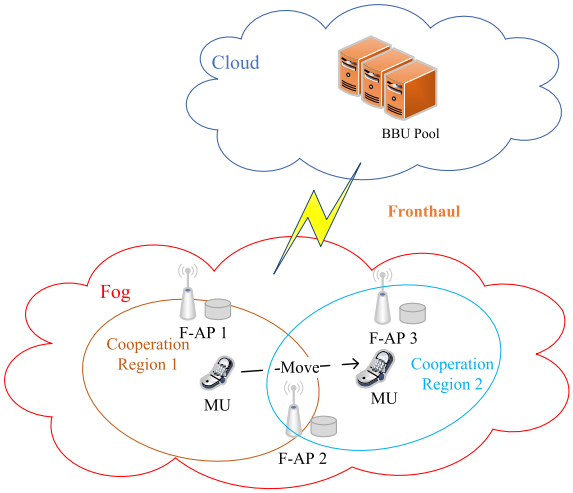


FIGURE 1. Cooperative content caching for mobile users in F-RAN.

III. SYSTEM MODEL

In this section, the cooperative content caching and delivery scenario for MUs is given first. Then, the user mobility in F-RAN system is described. Besides, the content caching and delivery processes are introduced respectively. Finally, this section formulates an average transmission delay minimization problem.

A. SYSTEM MODEL

The cooperative content caching and delivery scenario for mobile users is illustrated in Fig. 1. As shown in Fig. 1, an MU stays in the cooperation region of F-AP 1 and 2 at slot t, so as to download files from one of its associated F-APs (F-AP 1 and 2) nearby. When the MU moves to the cooperation region of F-AP 2 and 3 at slot t', its associated F-APs change to F-AP 2 and 3. This paper considers multiple cells scenario with the F-RAN architecture, which consists of M F-APs and K MUs. Let $\mathcal{F} = \{1, 2, \ldots, f, \ldots, M\}$ (and $\mathcal{U} = \{1, 2, \ldots, u, \ldots, K\}$) denote the F-AP set (and MU set), respectively. In the F-RAN system, the F-APs with limited storage capacity are deployed at the network edge, and the F-APs with close distance can cooperate and belong to the same region [12], which can be also called cooperation region. For simplification, it is assumed that each MU can be cooperatively served by two adjacent F-APs. For each MU, its associated F-APs set at time slot t is represented by $\mathcal{F}_u^t = \{f_{u,1}^t, f_{u,2}^t | f_{u,i}^t \in \mathcal{F}\}$.

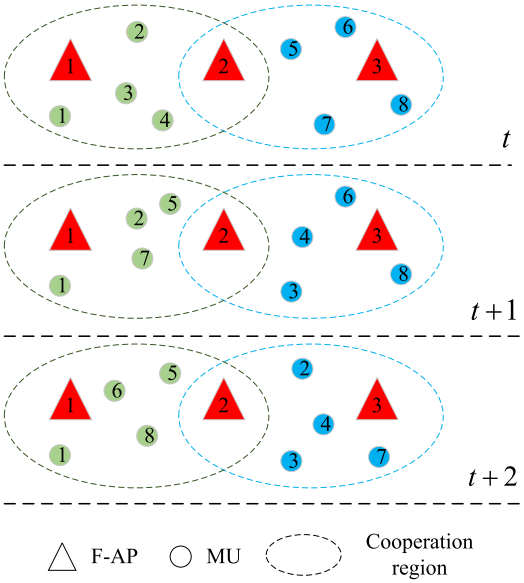


FIGURE 2. The time-varying topology relation between MUs and F-APs.

B. USER MOBILITY

The mobility pattern of MUs can be represented by the topology relation between MUs and F-APs. A $K \times M$ matrix $\mathcal{B}^t = [\beta^t_{u,f}]_{K\times M}$ is built to denote the topology relationship between MUs and F-APs at slot t, where each $\beta^t_{u,f}$ is a binary element, and is used to indicate the connectivity between MU u and F-AP f. If u lies within the coverage of f at slot t, $\beta^t_{u,f} = 1$, and $\beta^t_{u,f} = 0$ otherwise. The set of MUs in the coverage of f at slot t is defined as $\mathcal{U}^t_f = \{u \in \mathcal{U} | \beta^t_{u,f} = 1\}$. Likewise, the set of associated F-APs for u at slot t is defined as $\mathcal{F}^t_u = \{f \in \mathcal{F} | \beta^t_{u,f} = 1\}$.

Due to the user mobility, the relationship matrix $\mathcal{B}^t$ should be time-varying, which will seriously affect the caching policy. To model the various behaviors of MUs, $\mathcal{F}^t_u$ randomly varies every τ_u slots. In other words, MU u will stay in the cooperation region of $\mathcal{F}^t_u$ for τ_u slots. Taking Fig. 2 as an example for time-varying topology relation between MUs and F-APs. In the example, the triangles represent the F-APs, the circles denote the MUs, and the ellipses denote the cooperation regions. The circles with the same color means that they are located in the same cooperation region. The topology relation between MUs and F-APs is different during different time slot, since each MU can move to a random cooperation region after staying in a region for τ_u time slots. In other words, the dwell time and moving path for each MU may be distinct.

C. CONTENT CACHING

An MU requests the files from its associated F-APs. Since the files with different sizes are always divided into contents of the same size, it is assumed that all contents in the system have the same size S_c. As for an F-AP f, it can cache up to N_f contents from a content library $\mathcal{C} = \{1, 2, \ldots, c, \ldots, C\}$ in its local storage. Without loss of generality, assume $N_f \ll C$. For simplification, this paper assumes that each F-AP has the same storage capacity, i.e., $N_f = N$. A cache hit occurs when the requested content has been stored in the local storage. Otherwise, a cache miss occurs, and the requested content should be fetched from the remote content provider. Let a binary element $\mu^t_{c,f}$ denote the relationship between the requested content c and the storage of F-AP f, i.e., if c has been cached in the storage of f at slot t, $\mu^t_{c,f} = 1$,and $\mu^t_{c,f} = 0$ otherwise.

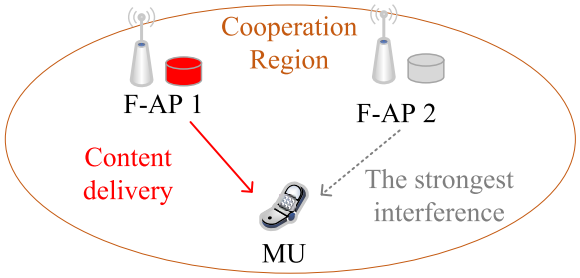


FIGURE 3. An illustration of content delivery.

In order to represent the content preferences of MU, a $K \times C$ matrix $\mathcal{P}^t = \{p^t_{u,c}\}_{K\times C}$ is built, where each $p^t_{u,c}$ is the probability that MU u requests content c at time slot t. For each MU, its preferences during each time slot are normalized such that $\sum_{c=1}^{C} p^t_{u,c} = 1$. It is assumed that different MUs have different content preferences. Specifically, content requests of each user u follow the Zipf distribution [37] with parameter κ_u. Therefore, the preference probability $p^t_{u,c}$ can be obtained by $\phi^t_u(c)^{-\kappa_u} / \sum_{c=1}^{C} c^{-\kappa_u}$, where $\phi^t_u(c) \in \{\phi^t_u(1), \phi^t_u(2), \ldots, \phi^t_u(C)\}$ is a random permutation of content library $\mathcal{C} = \{1, 2, \ldots, c, \ldots, C\}$.

Generally, cache hit ratio is a crucial indicator to evaluate the performance of a caching policy. In this work, the average cache hit ratio H_{av} during a long period T is defined as

$$H_{av} = \frac{\sum_t \sum_u \sum_c p^t_{u,c} \cdot \sigma^t_{u,c}}{T \cdot K}, \tag{1}$$

where T is the total number of time slots, and $\sigma^t_{u,c}$ is a binary element that indicates whether the content requested by MU has been stored in its associated F-APs at slot t. σ^t_c is given by

$$\sigma^t_{u,c} == \begin{cases} 0, & \sum_f \mu^t_{c,f} \cdot \beta^t_{u,f} < 1; \\ 1, & \sum_f \mu^t_{c,f} \cdot \beta^t_{u,f} \geq 1. \end{cases} \tag{2}$$

It is noticed that MU only downloads the requested content from one of its associated F-APs, though the requested content has been stored in more than one associated F-APs.

D. CONTENT DELIVERY

When an MU requests a content, the cache-hit content can be delivered from the local cache of its associated F-APs directly, and the cache-miss content should be fetched from

the remote content provider, which leads to extra transmission delay. In order to improve average hit ratio and reduce the average transmission delay, the caching policy should decide how to cache the contents at each slot. However, the delivery performance of requested contents depends not only on the caching policy, but also on the wireless channel states, e.g., channel fading, interference and so forth. For the cache-hit content, their transmission delay may be different, because of the different channel states.

As for the cache-hit content of MU u which is stored in F-AP f, the average transmission rate R_f^t during slot t is defined as

$$R_{f,u}^t = \mathbb{E}_{h_i^t}[B_u log_2(1 + \frac{p_0|h_i^t|^2\, l_i^{-\iota}}{n_0 B_u + p_0 \sum_{j\in\mathcal{F}_u^t\setminus i}|h_j^t|^2 l_j^{-\iota}})], \quad (3)$$

where $\mathbb{E}(\cdot)$ means the mathematical expectation, B_u is the transmission bandwidth for the MU u, p_0 is the transmit power of each F-AP, h_i^t is the small-scale fading channel, l_i is the distance between MU and the cache-hit F-AP, ι is the path loss factor, n_0 is the power spectral density of noise, and $p_0 \sum_{j\in\mathcal{F}_u^t\setminus i}|h_j^t|^2\, l_j^{-\iota}$ is the strongest interference power from the other associated F-AP. Since the strongest interference comes from the associated F-AP where the requested content has not been stored, assume that $l_i = l_j = l$. An illustration of content delivery is shown in Fig. 3. An MU dwells in the cooperation region of F-AP 1 and F-AP 2, and the cache-hit content of MU has been stored in F-AP 1. The MU downloads the requested content from F-AP 1, whilst the strongest interference comes from F-AP 2. Besides, it is assumed that the transmission bandwidth is allocated to each user equally.

The transmission delay of cache-hit content can be calculated by

$$d_{hit} = S_c / R_{f,u}^t. \quad (4)$$

As for a cache-miss content, its transmission delay is denoted by d_{miss}, which is higher than d_{hit}. Hence, the average transmission during a long period T is given by

$$D_{av} = \frac{\sum_t \sum_u \sum_c p_{u,c}^t \cdot [\sigma_{u,c}^t \cdot d_{hit} + (1 - \sigma_{u,c}^t) \cdot d_{miss}]}{T \cdot K}. \quad (5)$$

Since a file with a big size can be divided into several contents with a small size, assume that each requested content can be delivered within a single time slot. It means that the content delivery in each slot will not be interrupted by the movements of MUs. Consequently, S_c should be small enough to make $d_{hit} < d_{miss}$.

E. PROBLEM FORMULATION

Considering time-varying channel states, user mobility, diverse preferences of different MUs, limited cache capacity of each F-AP, this paper aims to find a cache update policy to minimize the average transmission delay, and the cooperative caching problem in F-RAN system is formulated as

$$\min\ D_{av}$$
$$s.t. \begin{cases} \sum_u \sum_c p_{u,c}^t \cdot \mu_{c,f}^t \cdot \beta_{u,f}^t \le N, & \forall f \in \mathcal{F} \quad (a); \\ \sum_u \sum_c p_{u,c}^t \cdot \sigma_{u,c}^t \le K, & \forall t \in \mathcal{T} \quad (b); \\ \sum_f \beta_{u,f}^t = 2, & \forall u \in \mathcal{U} \quad (c), \end{cases} \quad (6)$$

where constraint (6.a) means that each F-AP f is allowed to cache no more than N contents, constraint (6.b) indicates that the number of cache hits during each slot t is at most K, and constraint (6.c) represents that each MU u_k can be served by two N_p F-APs cooperatively.

IV. THE PROPOSED CACHE UPDATE POLICY

To figure out the caching problem raised in (6), a deep-Q-network with dueling architecture [25] is adopted. In this section, the cache update is modeled as an MDP [38]. Then, the workflow of dueling deep-Q-network is illustrated. Finally, the dueling DQN based cache update policy is proposed.

A. MARKOV DECISION PROCESS MODEL

A RL problem can be modeled as an MDP with state space $\mathcal{S}$, action space $\mathcal{A}$, transition probability P, reward function R and discount factor γ. In an MDP, the agent can learn how to interact with the environment to obtain the maximum average reward. In detail, the agent interacts with the environment in a sequence of discrete iteration steps. At each iteration step i, the agent observes the state s^i of the environment and chooses an action a^i. The agent will receive a reward $r^i = R(s^i, a^i)$ from the environment, after the selected action is executed. Then, the system transits into the next step $i+1$ with probability $P(s'|s,a) \triangleq \mathbb{P}[s^{i+1} = s'|s^i = s, a^i = a]$, where $\sum_{s'\in\mathcal{S}} P(s'|s,a) = 1$, for all $s \in \mathcal{S}, a \in \mathcal{A}$. Besides, a deterministic policy in an MDP is a mapping from state space $\mathcal{S}$ to action space $\mathcal{A}$, i.e.,$a = \pi(s)$. According to the Bellman equation, the average reward is defined as

$$\begin{aligned}\rho_\pi(s^0) &= \mathbb{E}[\sum_{i=0}^{\infty} \gamma^i R(s^i, \pi(s^i))|s^0] \\ &= \mathbb{E}[R(s^0, \pi(s^0)) + \gamma \sum_{s'\in S} P(s'|s^0, \pi(s^0))\rho_\pi(s')]. \end{aligned} \quad (7)$$

The goal of agent is to find a policy π^* to achieve the maximum average reward, that is

$$\pi^* = \arg\max_\pi \rho_\pi(s), \quad \forall s \in S. \quad (8)$$

Generally, although the problem above can be solved by dynamic programming, the curse of dimensionality occurs when the size of the problem is big. However, RL techniques, such as dueling DQN [25], DQN [39], etc., are applied as an effective approach to settle the problem without any priori knowledge of state transition probability P.

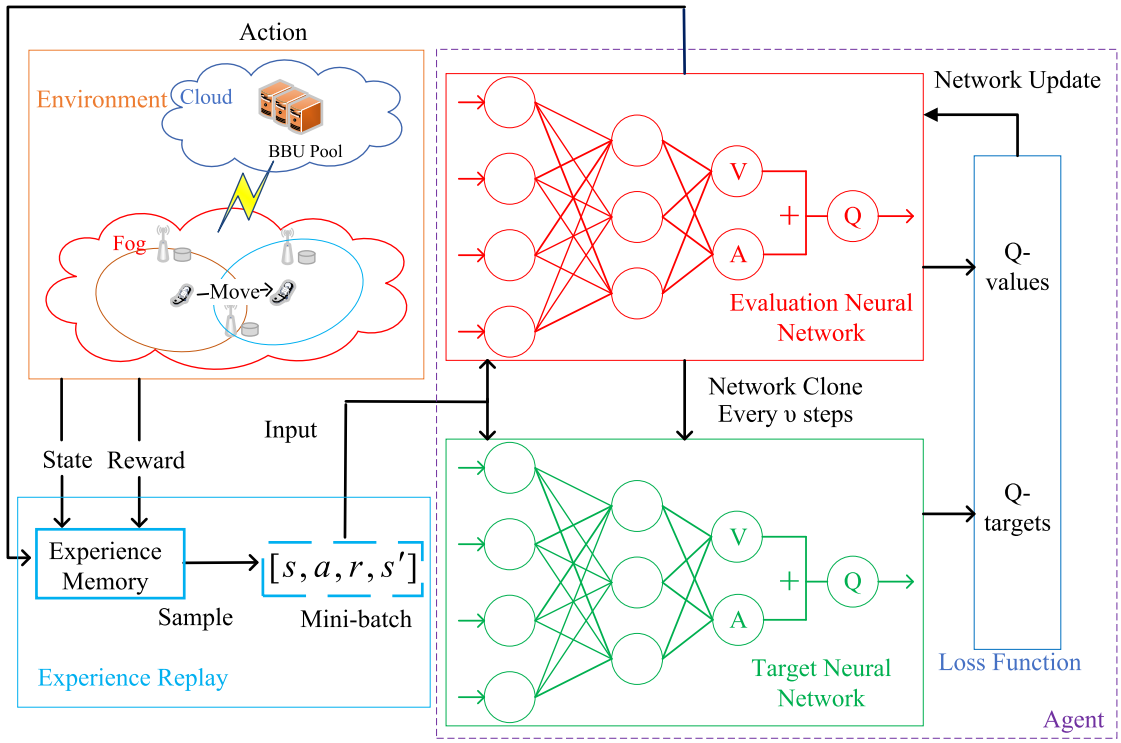


FIGURE 4. An illustration of dueling deep-Q-network.

B. WORKFLOW OF DUELING DEEP-Q-NETWORK

To employ the RL framework in this work, the state space $\mathcal{S}$, action space $\mathcal{A}$ and reward function R are defined as follows:

- **State space.** The state $s^i \in \mathcal{S}$ indicates the system information in each step i. Assume that each MU requests only one content during each slot. Then, the cache update procedure during each slot t can be divided into K states, and thus $i = t \cdot K + u$. Let $q_n^i = q_{u,n}^t$ denotes the total number of requests for the n^{th} content in the associated F-APs of MU u during last o slots at time t. The state s^i collects the information about cache status at each step i, and the cache status can be denoted by

$$s^i = [q_1^i, q_2^i, \ldots, q_n^i, \ldots, q_{2N_f}^i], \tag{9}$$

  where $2N_f$ is the total size of storages in the associated F-APs.
- **Action space.** The action $a^i \in \mathcal{A}$ represents the action that RL agent chooses at each step i. In order to limit the size of action space, the agent replaces only one cached content by the requested content or does nothing at each step i. Let $A^i = 0, 1, \ldots, n, \ldots, 2N_f$ denote all the candidate actions which can be chosen at step i, where $a^i = n(n \neq 0)$ means that the n^{th} cached content will be replaced, and $a^i = 0$ means that the requested content has been stored, so that the agent doesn't have to update the storage.
- **Reward function.** When the RL agent selects an action a^i under the state s^i, a reward r^i from the environment will be learned. To minimize the average transmission delay, the reward function is designed as

$$r^i = \frac{d_{miss} - d_{av}^i}{d_{miss}}, \tag{10}$$

  where d_{av}^i is the average transmission delay during each slot t, and d_{av}^i can be obtained by

$$d_{av}^i = \frac{1}{K} \sum_u \sum_c p_{u,c}^i \cdot [\sigma_{u,c}^i \cdot d_{hit} + (1 - \sigma_{u,c}^i) \cdot d_{miss}]. \tag{11}$$

  since d_{miss} is much higher than d_{av}^i in each step, r^i is always bigger than 0.

In nature DQN, neural network (NN) is employed to approximate a Q-value function which returns a Q-value for each input state-action pair (s, a). The Q-value $Q(s, a)$ is updated when the agent chooses an action a under the state s, and update function is defined as

$$Q(s, a) \leftarrow Q(s, a) + \alpha(\rho(s) + \gamma \max_{a'} Q(s', a') - Q(s, a)), \tag{12}$$

where $s' \in \mathcal{S}^{i+1}$ represents the next state, $a' \in \mathcal{A}^{i+1}$ denotes an action at next step. Factors α (and γ) denote learning rate (and reward decay) respectively. In addition, as shown in Fig. 4, target network and experience replay are employed to improve the learning efficiency of DQN framework [39]:

- **Target network.** In nature DQN, there are two separate NNs, evaluation NN and target NN. The evaluation NN is used to generate Q-values for given state-action pairs, and the target NN is utilized to generate Q-targets. The evaluation NN is constantly updated to make the

Q-values close to the Q-targets. The agent will reset the weights $\widehat{\boldsymbol{\theta}}$ of target NN by the weights $\boldsymbol{\theta}$ of evaluation NN every υ iteration steps.

- **Experience replay.** The agent can store the experiences $e^i = [s^i, a^i, r^i, s^{i+1}]$ in the experience memory D. Then, a mini batch of experiences are randomly sampled from the experience memory to update the evaluation NN.

When the agent updates the the evaluation NN, a loss function $L(\boldsymbol{\theta})$ will be adopted, and the loss function can be defined as

$$L(\theta) = (\widehat{Q}(s, a \mid \widehat{\boldsymbol{\theta}}) - Q(s, a \mid \boldsymbol{\theta}))^2, \tag{13}$$

where Q-targets $\widehat{Q}(s, a \mid \widehat{\boldsymbol{\theta}})$ can be obtained by

$$\widehat{Q}(s, a \mid \widehat{\boldsymbol{\theta}}) = r + \gamma \max_{a'} \widehat{Q}(s', a' \mid \widehat{\boldsymbol{\theta}}). \tag{14}$$

Then, the weights $\boldsymbol{\theta}$ can be obtained by minimizing the loss $L(\boldsymbol{\theta})$ via a gradient descent approach.

Furthermore, in dueling DQN, the Q-value function is divided into an advantage value $A(s, a)$ and a state value $V(s)$ to improve learning efficiency and accelerate convergence [25]. The Q-value function is given by

$$Q(s, a|\boldsymbol{\theta}) = A(s, a|\boldsymbol{\theta}) + V(s|\boldsymbol{\theta}). \tag{15}$$

Here, the advantage function is used to assess the value of the action that has been chosen, and the state value can measure the value of the state s. It is noticed that the state value is independent of actions. In practice, the agent can't distinguish $A(s, a)$ and $V(s)$. Since the agent can't obtain a unique solution to (15). To solve the unidentifiable problem above, the Q-value function can be calculated as

$$Q(s, a|\boldsymbol{\theta}) = V(s|\boldsymbol{\theta}) + (A(s, a|\boldsymbol{\theta}) - \frac{1}{|A|}\sum_{a'} A(s, a'|\boldsymbol{\theta})). \tag{16}$$

C. DUELING DEEP-Q-NETWORK BASED CACHE UPDATE POLICY

The proposed dueling DQN based cache update policy is illustrated in Algorithm 1. In the background, the RL agent can collect information including cache status, transmission delay, requested contents and so forth, and the dueling DQN will be trained for N_{ep} episodes. When the agent is well trained, the weights of NN will be stored and utilized for cache update. Note that the agent use greedy policy to explore new policies in the training phase, and the factor ϵ is set to 1 in the testing phase.

Algorithm 1 Dueling DQN Based Cache Update Policy

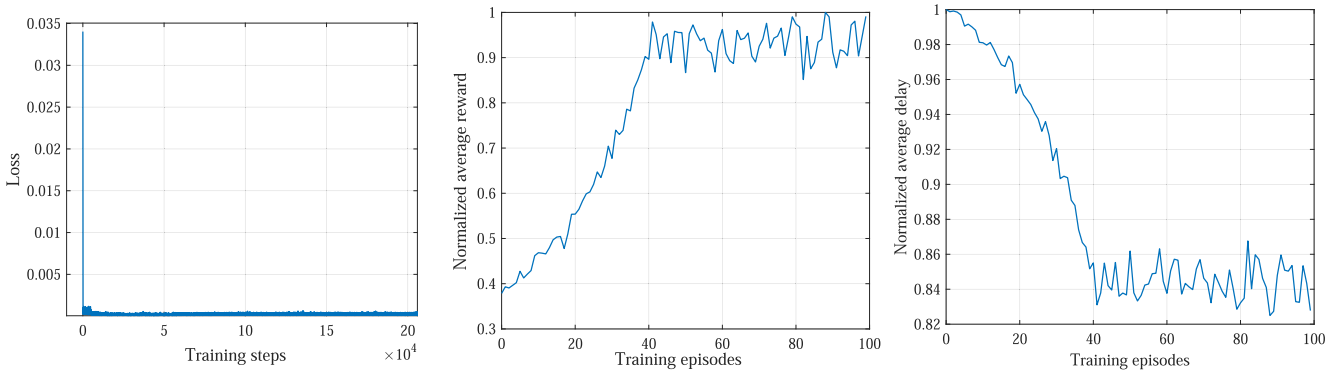
Randomly initialize an evaluation NN $Q(s, a \mid \boldsymbol{\theta})$ with weights $\boldsymbol{\theta}$ and a target NN $\widehat{Q}(s, a \mid \widehat{\boldsymbol{\theta}})$ with weights $\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}$.
Initialize a experience memory D with a size of N_D.
for each episode $ep \in [1, N_{ep}]$ **do**
  for each $t \in [1, T]$ **do**
    for each $u \in [1, K]$ **do**
      MU u requests a content c_u^t.
      if the content c_u^t has been stored in the associated F-APs $\mathcal{F}_u^t$ **then**
        Download the content from the local cache.
      else
        if the storage of $\mathcal{F}_u^t$ is not full **then**
          Fetch the content from the remote content provider.
          Cache the currently requested content c_u^t in the local cache.
        else
          Observe the system state $s^i (i = t \cdot K + u)$.
          Choose an action $a^i = \arg\max_a Q(s, a)$ with probability ϵ, or a random action with probability $1 - \epsilon$.
          Replace the n^{th} stored content in the storage of $\mathcal{F}_u^t$ with c_u^t.
          Receive the reward r^i.
          Store the experience $e^i = [s^i, a^i, r^i, s^{i+1}]$ in D.
          Randomly sample a mini batch of experiences from D.
          if episode terminates at step i **then**
            Set $y^i = r^i + \gamma \max_{a'} \widehat{Q}(s', a' \mid \widehat{\boldsymbol{\theta}})$
          else
            $y^i = r^i$
          end if
          Update $\boldsymbol{\theta}$ by minimizing the loss $(y^i - Q(s, a \mid \boldsymbol{\theta}))^2$ via a gradient descent algorithm.
          Reset the target NN $\widehat{Q}$ every υ steps by replacing weights $\widehat{\boldsymbol{\theta}}$ with $\boldsymbol{\theta}$.
        end if
      end if
    end for
  end for
end for

V. SIMULATION RESULTS AND PERFORMANCE EVALUATION

In this section, simulations are preformed to validate the performance of the proposed caching policy. Firstly, the simulation parameters are given. Then, the convergence of dueling DQN is analyzed. Moreover, compared with FIFO, LFU and LRU caching policies, the performance of the proposed caching policy is evaluated in terms of average hit ratio and average transmission delay.

A. SIMULATION SETUP

In the simulations, this paper considers an F-RAN with M F-APs and K MUs. The preference of each MU is distinct, and the content requests of each MU follow the Zipf distribution with parameter $\kappa_u = 1.1$. The small-scale channel gain $|h^t|^2$ follows exponential distribution. Each MU in the cooperation region is served by two F-APs. Each MU stays in a cooperation region for τ_u time slots. In other words, elements in the



(a) Loss between Q-values and target values for varying training steps. (b) Normalized average reward for varying episodes. (c) Normalized average delay for varying episodes.

FIGURE 5. Learning curves of the proposed dueling DQN based caching policy ($M = 5, K = 10, N = 15$).

u^{th} row of topology relationship matrix $\mathcal{B}^t$ are regenerated every τ_u slots. When an MU dwells in a cooperation region, the distance between the MU and its associated F-APs is l. For simplification, assume that the coverage distance is a constant value $l = 100$ m. Besides, the system bandwidth is set to 20MHz, and allocated to each MU equally. Some main parameters are listed in Table 1. In the simulation, the training set of dueling DQN is generated by a random seed, and the testing set are generated by another five random seeds.

To validate the performance of the proposed caching policy, the simulation results are compared with following caching policies:

- **First in first out (FIFO).** If the currently requested content hasn't been stored in the local cache, FIFO policy will replace the content which is stored earliest by the new content.
- **Least recently used(LRU).** When the LRU policy updates the local cache, the stored content that is least recently requested will be replaced by the new content.
- **Least frequently used (LFU).** The LFU policy records the number of requests for each stored content. The stored content with the least requests number will be replaced.

B. CONVERGENCE ANALYSIS

Fig. 5 illustrates the learning curves of the proposed dueling DQN based caching policy for loss, normalized average reward and normalized average transmission delay.

Fig. 5(a) shows the loss between the target values and the Q-values for varying training steps. From the figure, the loss curve descends quickly, as the increase of training steps. With enough training steps, the loss converges to a stable state. Fig. 5(b) presents the average reward of each episode. As the increase of episodes, the average reward gradually rises. On the contrary, the average transmission delay decreases piece by piece. Note that average reward and average transmission delay start to fluctuate when N_{ep} is about 60, since the maximum value of greedy factor ϵ in training phase is set to 0.9, so that RL agent may choose a suboptimal or even bad action with probability 0.1.

TABLE 1. Simulation Parameters.

Number of F-APs (M)	5
Number of UEs (K)	5,10,15,20,25
Length of period $(\mathcal{T})$	1000 slots
Size of each content (S_c)	10Mbit
Transmission bandwidth for each MU (B_u)	$20/K$ MHz
Dwell time (τ_u)	$10 \sim 100$ slots
Transmission power (p_0)	46 dBm
Noise power spectral density (n_0)	-174 dBm/Hz
Coverage radius (l)	100 m
Path loss factor (ι)	4
Transmission delay of cache-miss content (d_{miss})	20 s
Size of storage in F-AP (N_f)	5,10,15,20,25
length of observation period (o)	$5 \cdot N$
Learning rate (α)	0.01
Reward decay(γ)	0.9
Maximum greedy factor (ϵ)	0.9
Number of episodes (N_{ep})	100
Size of memory buffer (N_D)	20000
Size of mini batch	32
Memory replay period (υ)	4 iteration steps
Length of memory replay period	5000
Number of hidden layers	2
Number of nodes per layer	$4 \cdot N$

C. PERFORMANCE EVALUATION

In comparison of FIFO, LRU and LFU caching policies, the proposed caching policy is validated in terms of average hit ratio and average transmission delay.

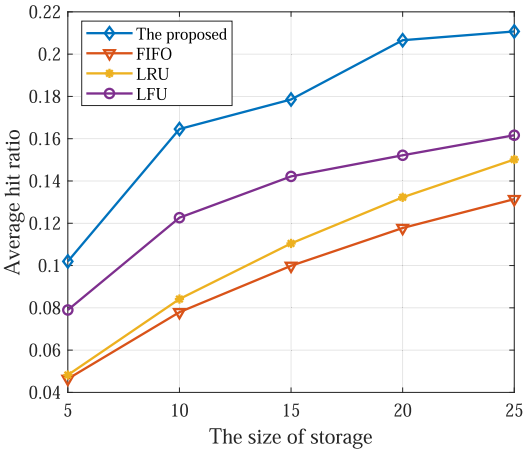


FIGURE 6. Average hit ratios of different caching policies for varying storage sizes ($M = 5, K = 10, \tau_u = 10$).

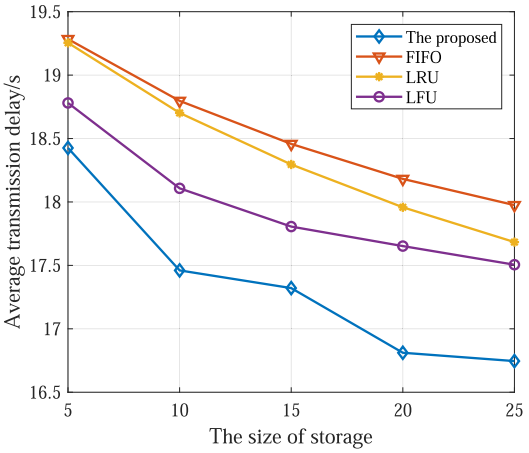


FIGURE 7. Average transmission delays of different caching policies for varying storage sizes ($M = 5, K = 10, \tau_u = 10$).

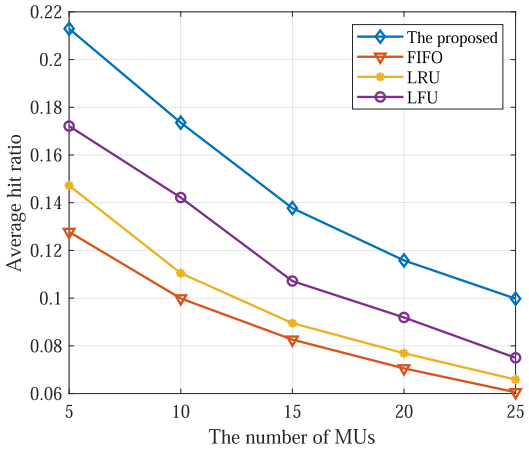


FIGURE 8. Average hit ratios of different caching policies for varying MU numbers ($M = 5, N = 15, \tau_u = 10$).

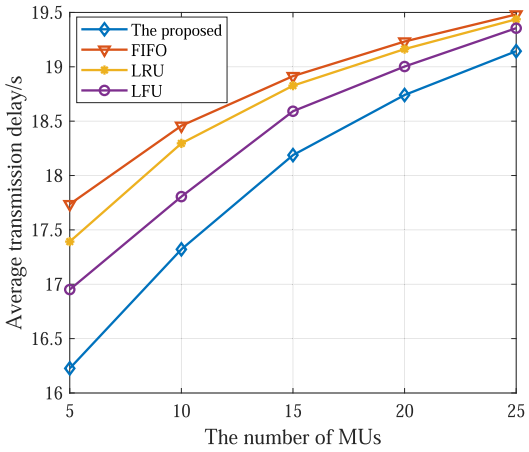


FIGURE 9. Average transmission delays of different caching policies for varying MU numbers($M = 5, N = 15, \tau_u = 10$).

To evaluated the influences of storage size in average hit ratio and average transmission delay, simulations are performed for varying storage sizes. In the simulations, the number of F-APs M are set to 5, the number of MUs K is set to 10, the dwell time of each MU τ_u is 10 slots, and the storage size N varies in a range of [5, 25]. The average hit ratios of different caching policies for varying storage sizes are shown in Fig. 6. From the figure, all the average hit ratios of different caching policies rise, as the increase of storage size. For each storage size, the average hit ratio of the proposed caching policy is higher than those of other policies. Fig. 7 illustrates the average transmission delays of different caching policies for varying storage sizes. In the figure, all the average transmission delays of different caching policies descend, as the storage size expands. Besides, the average transmission delay of the proposed caching policy is much better than those of other policies. This is because the larger storage can store more requested contents, so that the MUs can download more cache-hit contents from local cache directly instead of fetching the cache-miss contents from remote server.

The influences of MU number in average hit ratio and average transmission delay is also confirmed, and the simulation results are depicted in Fig. 8 and Fig. 9. In the figures, the number of F-APs M is 5, the size of storage in each F-AP N is 15, the dwell time of each MU τ_u is 10 slots, and the number of MUs K varies from 5 to 25. From Fig. 8, it can be seen that all the average hit ratios of different caching policies decrease gradually, as the increase of MU number. For each K, the average hit ratio of the proposed caching policy is higher than those of other policies. From Fig. 9, the average transmission delays of different caching policies go up, as K increases. Moreover, the average transmission delay of the proposed caching policy is lower than those of other policies for different MU numbers. Since the preferences of different MU are different, the kinds of the requested contents increases, as the increase of MU number. Consequently, the number of cache-hit contents decreases, if the number of MUs increases and the storage size remains unchanged.

Note that the dwell times of different MUs are set the same value in Fig. 6∼9. Actually, the dwell times of different MUs may be different because of their

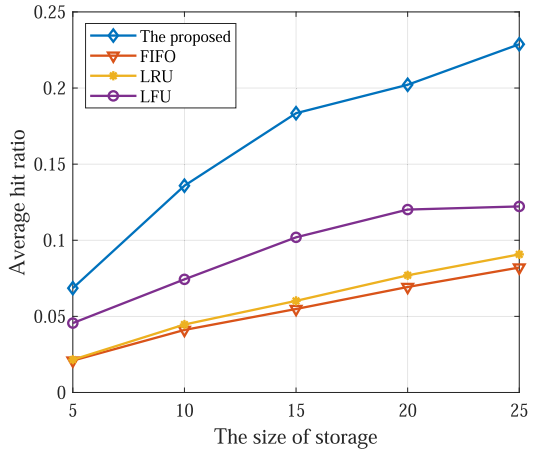


FIGURE 10. Average hit ratios of different caching policies for varying storage sizes, and the dwell times of different MUs are different($M = 5, K = 10$).

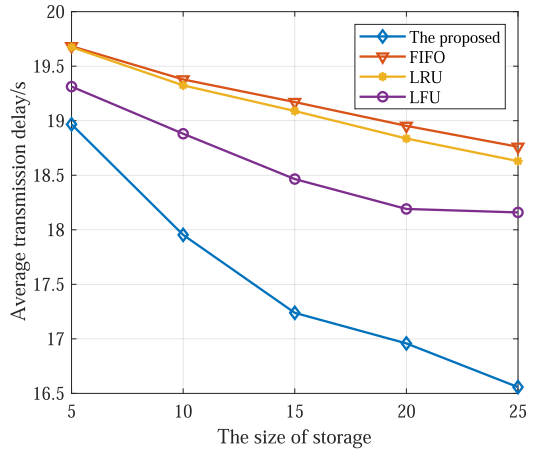


FIGURE 11. Average transmission delays of different caching policies for varying storage sizes, and the dwell times of different MUs are different ($M = 5, K = 10$).

random behaviors. Accordingly, the dwell times of 10 MUs are set to {10, 20, 30, 40, 50, 60, 70, 80, 90, 100} slots in Fig. 10 and Fig. 11 to simulate the unpredictable user mobility. Besides, the number of F-APs M are set to 5, the number of MUs K is set to 10. As the storage size increases, the average hit ratios raise, while the average transmission delays drop. Since each MU moves randomly, the stored contents may not be requested at next slot. Although the movements of MUs are arbitrary and ruleless, the dueling DQN can still work well. Furthermore, the proposed caching policy provides superior average hit ratio and average transmission delay, compared to other traditional policies.

From the simulation results, the following conclusions can be summarized.

- As the increase of the storage size, the average hit ratios of caching policies ascent, while the average transmission delays of caching policies descent.
- As the number of MU raises, the average hit ratios of caching policies fall, whilst the average transmission delays of caching policies go up.
- The proposed caching policy can work well in various scenarios with different storage sizes, user densities and mobility patterns. Furthermore, the proposed caching policy outperforms other traditional caching policies in different scenarios.

VI. CONCLUSION

In this work, a cache update problem in F-RAN is investigated, by taking into account diverse user preferences, random user mobility, time-varying channel fading and cooperation between adjacent F-APs. Resorting to the dueling DQN technique, this paper develops a delay-aware cache update policy for MUs in F-RAN. In the proposed dueling DQN based caching policy, the average transmission delay of MUs is designed as the reward at each iteration step to achieve the minimum average transmission delay. In order to analyze performance of the proposed caching policy, simulations are performed in various scenarios with different storage sizes, user densities and mobility patterns, compared with three traditional caching policies, i.e., FIFO, LRU and LFU. The simulation results show that the proposed caching policy can not only improve the average hit ratio, but also reduce the average transmission delay. Although the number of MUs becomes denser and the movements of MUs are arbitrary and ruleless, the proposed cache update policy can still show much more superiority than other caching algorithms.

Although the caching problem studied in this paper is under F-RANs, the proposed caching policy can still work in other network scenarios, e.g. mobile edge computing systems. It is noticed that the transmission bandwidth is allocated to each user equally in this paper. Obviously, it is not an efficient way to make use of radio resource because of the time-varying and diverse user demands. However, a radio resource efficient cache update policy will be investigated in future works.

REFERENCES

[1] Cisco, San Jose, CA, USA. (Feb. 2019). *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html
[2] P. Zhang, X. Kang, D. Wu, and R. Wang, "High-accuracy entity state prediction method based on deep belief network toward IoT search," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 492–495, Apr. 2019.
[3] D. Wu, Z. Zhang, S. Wu, J. Yang, and R. Wang, "Biologically inspired resource allocation for network slices in 5G-enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9266–9279, Dec. 2019.
[4] Z. Li, Y. Jiang, Y. Gao, L. Sang, and D. Yang, "On buffer-constrained throughput of a wireless-powered communication system," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 2, pp. 283–297, Feb. 2019.
[5] P. Zhang, X. Kang, X. Li, Y. Liu, D. Wu, and R. Wang, "Overlapping community deep exploring-based relay selection method toward multi-hop D2D communication," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1357–1360, Oct. 2019.
[6] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul. 2016.
[7] X. Zhang and Q. Zhu, "Hierarchical caching for statistical QoS guaranteed multimedia transmissions over 5G edge computing mobile wireless networks," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 12–20, Jun. 2018.
[8] T. Zhang, X. Xu, L. Zhou, X. Jiang, and J. Loo, "Cache space efficient caching scheme for content-centric mobile ad hoc networks," *IEEE Syst. J.*, vol. 13, no. 1, pp. 530–541, Mar. 2019.

[9] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, "QoE-driven mobile edge caching placement for adaptive video streaming," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 965–984, Apr. 2018.

[10] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, and X. Shen, "Content popularity prediction towards location-aware mobile edge caching," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 915–929, Apr. 2019.

[11] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, D. Niyato, and D. I. Kim, "Distributed deep learning at the edge: A novel proactive and cooperative caching framework for mobile edge networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1220–1223, Aug. 2019.

[12] Y. Jiang, M. Ma, M. Bennis, F.-C. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1268–1283, Feb. 2019.

[13] B. Chen and C. Yang, "Caching policy for cache-enabled D2D communications by learning user preference," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6586–6601, Dec. 2018.

[14] P. Cheng, C. Ma, M. Ding, Y. Hu, Z. Lin, Y. Li, and B. Vucetic, "Localized small cell caching: A machine learning approach based on rating data," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1663–1676, Feb. 2019.

[15] C. Zhong, M. C. Gursoy, and S. Velipasalar, "A deep reinforcement learning-based framework for content caching," in *Proc. 52nd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2018, pp. 1–6.

[16] W. Jiang, G. Feng, S. Qin, T. S. P. Yum, and G. Cao, "Multi-agent reinforcement learning for efficient content caching in mobile D2D networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1610–1622, Mar. 2019.

[17] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 1, pp. 180–190, Feb. 2018.

[18] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 80–87, Jun. 2018.

[19] P. Lin, K. S. Khan, Q. Song, and A. Jamalipour, "Caching in heterogeneous ultradense 5G networks: A comprehensive cooperation approach," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 22–32, Jun. 2019.

[20] Y. Zhou, Z. Zhao, R. Li, H. Zhang, and Y. Louet, "Cooperation-based probabilistic caching strategy in clustered cellular networks," *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 2029–2032, Sep. 2017.

[21] D. Wu, L. Zhou, Y. Cai, and Y. Qian, "Collaborative caching and matching for D2D content sharing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 43–49, Jun. 2018.

[22] D. Wu, Q. Liu, H. Wang, Q. Yang, and R. Wang, "Cache less for more: Exploiting cooperative video caching and delivery in D2D communications," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1788–1798, Jul. 2019.

[23] B. Guo, X. Zhang, Y. Wang, and H. Yang, "Deep-Q-network-based multimedia multi-service QoS optimization for mobile edge computing systems," *IEEE Access*, vol. 7, pp. 160961–160972, 2019.

[24] D. Wu, H. Shi, H. Wang, R. Wang, and H. Fang, "A feature-based learning system for Internet of Things applications," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1928–1937, Apr. 2019.

[25] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," 2015, *arXiv:1511.06581*. [Online]. Available: https://arxiv.org/abs/1511.06581

[26] Y. Li, C. Zhong, M. C. Gursoy, and S. Velipasalar, "Learning-based delay-aware caching in wireless D2D caching networks," *IEEE Access*, vol. 6, pp. 77250–77264, 2018.

[27] Z. Zhang and L. Wang, "Social tie-driven content priority scheme for D2D communications," *Inf. Sci.*, vol. 480, pp. 160–173, Apr. 2019.

[28] T. Dang and M. Peng, "Joint radio communication, caching, and computing design for mobile virtual reality delivery in fog radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1594–1607, Jul. 2019.

[29] Z. Li, J. Chen, and Z. Zhang, "Socially aware caching in D2D enabled fog radio access networks," *IEEE Access*, vol. 7, pp. 84293–84303, 2019.

[30] Z. Zheng, L. Song, Z. Han, G. Y. Li, and H. V. Poor, "A Stackelberg game approach to proactive caching in large-scale mobile edge networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5198–5211, Aug. 2018.

[31] J. Jiao, X. Hong, and J. Shi, "Proactive content delivery for vehicles over cellular networks: The fundamental benefits of computing and caching," *China Commun.*, vol. 15, no. 7, pp. 88–97, Jul. 2018.

[32] X. Zhang, T. Lv, Y. Ren, W. Ni, N. C. Beaulieu, and Y. J. Guo, "Economical caching for scalable videos in cache-enabled heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1608–1621, Jul. 2019.

[33] O. Ayoub, F. Musumeci, M. Tornatore, and A. Pattavina, "Energy-efficient video-on-demand content caching and distribution in metro area networks," *IEEE J. Sel. Areas Commun.*, vol. 3, no. 1, pp. 159–169, Mar. 2019.

[34] S. O. Somuyiwa, A. Gyorgy, and D. Gunduz, "A reinforcement-learning approach to proactive caching in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1331–1344, Jun. 2018.

[35] L. Hou, L. Lei, K. Zheng, and X. Wang, "A *Q*-learning-based proactive caching strategy for non-safety related services in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4512–4520, Jun. 2019.

[36] Z. Zhang, Y. Yang, M. Hua, C. Li, Y. Huang, and L. Yang, "Proactive caching for vehicular multi-view 3D video streaming via deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 18, no. 5, pp. 2693–2706, May 2019.

[37] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, vol. 1, Mar. 1999, pp. 126–134.

[38] R. Bellman, "A Markovian decision process," *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679–684, 1957.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

BOREN GUO received the B.Eng. degree in electronic information engineering from the Beijing University of Posts and Telecommunications (BUPT), in 2015, where he is currently pursuing the Ph.D. degree with the Wireless Theories and Technologies Laboratory. His research interests include C-RAN, F-RAN, MEC, deep reinforcement learning, and other 5G NR techniques.

XIN ZHANG received the B.Eng. degree in communications engineering, the M.Eng. degree in signal and information processing, and the Ph.D. degree in communications and information systems from the Beijing University of Posts and Telecommunications (BUPT), in 1997, 2000, and 2003, respectively. He joined BUPT, in 2003, currently working with the Wireless Theories and Technologies Laboratory as an Associate Professor, and focuses the research mainly on key technologies and performance analysis of air interface of wireless networks.

QIWEI SHENG received the B.Eng. degree in communication engineering from the Beijing University of Posts and Telecommunications, in 2018, where he is currently pursuing the M.Eng. degree. His research interests include deep reinforcement learning, C-RAN, MEC, and other 5G technologies.

HONGWEN YANG received the B.S. and M.S. degrees from the Beijing University of Posts and Telecommunications (BUPT), in 1984 and 1987, respectively. After graduating, he joined the Faculty of BUPT, where he is currently a Professor. His research mainly focuses on wireless physical layer, including modulation and coding, MIMO, and OFDM.