MDC^2: An Integrated Communication and Computing Framework to Optimize Edge-assisted Caching for Improved Multimedia Services in UAV-based IoT Networks

Lujie Zhong, Shujie Yang, Kefei Song, Mu Wang, Ke Jiang, and Gabriel-Miro Muntean, *Fellow, IEEE*

***Abstract*—Multi-access Edge Computing (MEC) has revolutionized the delivery of large-scale mobile multimedia services by endowing network edge with computing and caching capabilities. This not only relieves the load on core networks, but also significantly reduces data access latency. However, deploying edge data centers with a high density to accommodate the growing demand for multimedia services is not cost-effective. With the rapid development of the Internet of Things (IoT) industry, recent studies have shown that by allowing UAVs with integrated computing and communication to form a Mobile Device Cloud (MDC) environment via UAV-to-UAV (U2U) communications in IoT networks, UAVs can play an important role in assisting cellular networks with multimedia delivery and providing excellent service for IoT devices on the ground. While a MDC environment composed of UAVs offers flexibility and cost-effectiveness, the challenge remains in allocating caching resources in a timely manner to meet the dynamic content demands. To address this challenge, we design a novel Mobile Device Cloud-enabled Caching (MDC^2) framework, which makes use of the available caching and U2U communication capabilities to enable any UAV to obtain dynamically content from other nearby UAVs via the IoT network. By modeling the dynamic network status as a fluid-based system, MDC^2 employs a dynamic caching allocation algorithm to minimize both service latency and caching costs. Extensive experiments demonstrate that MDC^2 outperforms a state-of-the-art MDC multimedia delivery approach by improving average cache utilization with over 40% and reducing average access latency with more than 25%.**

***Index Terms*—UAV, Internet of Things (IoT), Caching, Mobile Device Cloud, Multimedia Streaming**

I. INTRODUCTION

THE latest estimates predict that the virtual reality-related network traffic will increase more than 20 times and the Internet video traffic with more than 25% in the next five years [1]. Multi-access Edge Computing (MEC), also known as Mobile Edge Computing, is seen as a promising solution for delivering large-scale multimedia services as it allows for network edges to possess and make available for third party usage significant computational power and caching resources [2]–[4]. However, the deployment costs of MEC are extremely high, making it increasingly challenging to meet the growing demands of rich multimedia services in the future 5G and beyond networks [5].

Recently, with the rapid development of UAVs with integrated computing and communications support, the deployment of UAVs to assist ground base stations in content distribution over IoT networks has attracted extensive attention [6], [7], especially in Mobile Device Cloud (MDC) scenarios [8], [9]. In such MDC contexts, each UAV caches and distributes contents via UAV-to-UAV (U2U) transmissions in the IoT network by utilizing its own caching resources. This enables a highly flexible enhancement of the capability of the edge cloud and supports scalability. The MDC environment composed of UAVs in an IoT network context is becoming increasingly popular as it provides a cost-effective alternative to MEC, offering a resource-rich environment and general management frameworks. At the same time, the low cost of UAVs, convenience of server deployment and flexibility of mobility make them play an important role in optimizing the performance of cellular network spectrum efficiency, transmission delay and throughput of multimedia services. However, there also exist certain challenges which need to be investigated before any large-scale deployment. First, different from data centers, the availability of caching resources for caching multimedia content at each UAV is highly dynamic. Secondly, UAVs have their own network status, caching, and energy resources for caching; these resources may get exhausted when content is aggressively cached. Moreover, as UAVs move continuously, the limited range of U2U communications and fluctuations of the wireless connections result in discontinuous multimedia content delivery. External non-stationary aspects make reliable delivery of latency-sensitive multimedia services very challenging in MDC environments.

This paper primarily focuses on addressing these MDC challenges by making the following contributions:

- We design a novel Mobile Device Cloud-enabled Caching framework (MDC^2), which uses the available caching and U2U communication capabilities at any UAV to obtain content from other nearby UAVs via the IoT network. We present an online optimization model for caching config-

Corresponding author: Shujie Yang.

L. Zhong is with Information Engineering College, Capital Normal University, Beijing, China and Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education. (email: zhonglj@cnu.edu.cn)

S. Yang, K. Song, M. Wang, and K. Jiang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. (email: sjyang@bupt.edu.cn, kfsoong@163.com, muwang@bupt.edu.cn, jke@bupt.edu.cn). Shujie Yang is the co-first author and has contributed to this work equally with Lujie Zhong.

G.-M. Muntean is with the Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Dublin 9, Ireland (e-mail: gabriel.muntean@dcu.ie).

uration in MDC environments and the MDC^2 algorithm.

- Considering the changing mobility characteristics, we capture content changes based on video requirements and design a novel algorithm based on fluid models. The algorithm contains three components (a) a fluid-based model for estimating the resource demand for different contents at different time slots, (b) a caching algorithm for configuring the caching decisions in the network, and (c) a smooth playback-ensured fast delivery policy (SPFDP) for selecting the providers with better delivery performance to requesters.
- In contrast to the classical caching methods, which allocate the caching resources statically, MDC^2 evaluates the number of UAVs in various states during each time period and aims to minimize the peak number of clients requesting content and the amount of copies required. Also, the MDC^2 algorithm is such designed to optimize content request delay and caching configuration. The results of various experiments show that compared to existing caching algorithms, MDC^2 performs better by reducing the delay and improving cache utilization.

The rest of the paper is organized as follows. Section II discusses the recent related works and highlights the novelty of the proposed framework. Sections III and IV focus on the system design and problem formulation. Section V introduces the novel caching configuration algorithm. Section VI assesses the performance of the proposed approach through extensive experiments and Section VII draws conclusions.

II. Related Works

Recently, there has been a lot of research aimed at improving edge caching capabilities through static deployment of UAVs in IoT networked systems. For example, in the context of limiting the energy consumption of UAVs, Cheng *et al.* proposed a joint optimization problem of content popularity and cache placement, and obtained the minimum content acquisition delay by solving two two-dimensional matching problems [10]. Wang *et al.* constructed a mathematical model based on the temporal and spatial distribution characteristics of UAV network content popularity, residual cache changes and user download experience effects, and used this mathematical model to obtain an optimized content placement method in an UAV-based network [11]. Zeng *et al.* proposed a user association optimization algorithm based on penalty successive convex approximation and an UAV deployment and hierarchical cache placement algorithm based on penalty difference-of-convex programming. The proposal minimizes the total video access latency for all users through a combination of layered cache placements and UAV deployments, as well as user-associated design [12]. Luo *et al.* formulated a joint optimization problem for UAV deployments and content placements by considering heterogeneous user activity levels and dynamic content libraries. The average request delay is minimized by solving two subproblems. Specifically, they used a weighted K-means method for UAV deployment and a Q-learning algorithm to learn best content placement [13].

Fan *et al.* proposed a traffic load balancing scheme in UAV-assisted fog networks to minimize wireless latency for networked users. In this scheme, the authors divided the problem into two sub-problems and designed two algorithms to optimize UAV placement and user association, respectively [14]. Fazele *et al.* proposed an iterative algorithm that aims to jointly optimize the number of UAVs, their 3D placement, and the cache placement probability of content stored in UAVs and IMDs by maximizing secure cache throughput [15]. Fan *et al.* proposed an emergency communication network framework for UAVs that supports caching, and designed a content-centric association strategy within a specified cyclic cache region, in which each UAV is equipped with a cache unit that adopts a probabilistic cache strategy [16]. Zhuang *et al.* provided a popularity-based information mining and content delivery strategy for blind popularity-distribution in D2D scenarios. The authors designed a multi-armed bandit model and proposed single-cache and multi-cache placement strategies based on online learning to achieve blind popularity in D2D networks [17]. Most of these caching schemes depend on probability-based estimates of content popularity. However, content popularity is difficult to predict and can change frequently, particularly in mobile environments where demands are highly variable. To address this critical limitation of existing caching solutions, our approach involves regularly updating the popularity probability of the content through a fluid-based state transition model.

Some recent studies also utilize reinforcement learning algorithms for cache configuration. In [18], Somesula *et al.* proposed a deep reinforcement learning based cooperative caching mechanism to implement efficient caching in D2D enabled MEN. Zhang *et al.* studied cooperative content delivery from various cache-enabled network edges [19]. By jointly considering cache and cooperative delivery, the researchers formulated a finite-time Markov Decision Process (MDP) and proposed an algorithm to minimize the average delay. In [20], Zong *et al.* presented the challenge of cache management under dynamic content popularity as a forecasting issue and created a framework, which uses a deep reinforcement learning ensemble approach, as a solution. In [21], Somesula *et al.* modeled the cooperative cache update problem as a partially observable MDP (POMDP) problem and designed a multi-agent recurrent cooperative caching algorithm to maximize the cumulative reward. However, network condition can change dynamically, complicating much the state space in reinforcement learning (RL) models. Also, the generalization problem can be hard to solve and RL-based models may not satisfy diverse and differentiated users' requests.

Additionally, UAVs and users are not stationary, and their locations changes much over time. Most of the above solutions deploy UAVs dynamically or statically according to the characteristics of the environment and the distribution characteristics of the population, and solutions may not be universal. Our scheme describes the motion of UAVs as a random waypoint model. By solving the dynamic distribution problem of cache under this motion model, our algorithm has stronger generalization performance and is expected to show good performance in different scenarios. In addition, in order to address the dynamic changes in the demand for multimedia resources generated by the mobility of IoT

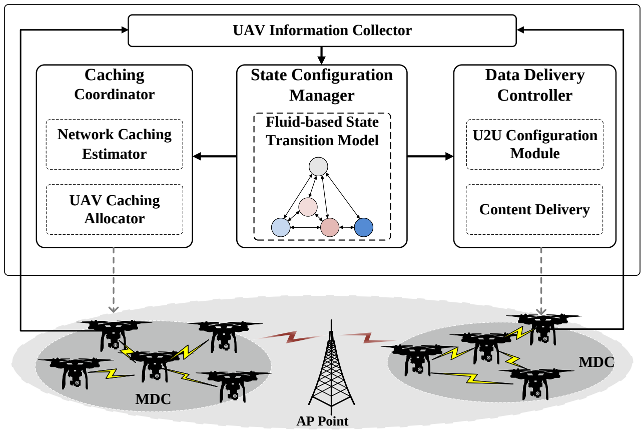


Fig. 1. Illustration of the proposed MDC^2 framework

devices on the ground, our approach redefines the state of each UAV throughout the content distribution process in MDC environments via an IoT network. In this work, five innovative state definitions are proposed and all possible state transitions in MDC environments are covered. The system architecture, including state definitions and transitions are described next.

III. System Architecture

A. MDC^2 Control Logic

Fig. 1 presents the proposed system architecture. The framework consists mainly of the following modules:

- **State Configuration Manager**: This module maintains states based on the locations of the UAVs with respect to the Access Point (AP)[1]. The possible states are *initial*, *offline*, *pending*, *transmission* and *caching* state. For each UAV, states can change over time, which will be further elaborated in the following sections.
- **Caching Coordinator**: This module allocates the caching resources to different video content based on a caching scheme. There is one caching element in each UAV that collects the video contents from the corresponding AP.
- **Data Delivery Controller**: This controller operates at each AP and receives video requests and selects the video provider for each IoT device. This module also manages the main handover of UAVs between APs. It is based on controller-to-controller communication, which is perceived as a message exchange such that the current controller de-registers the UAV and sends a "$Connect\_To$" message to the destination controller. An $< IP, MAC >$ tuple is stored in the local database and is carried forward to the destination controller. We divide the area in clusters such as each cluster has one controller and all APs that are communicating in this area will have one aggregated controller location. When an AP outside of a cluster responds with an "$Accept$" message, then the controller has completed the handover procedure.

  If a drop occurs due to an unexpected signal strength decrease or any other misbehavior, then the handover fails. This paper does not consider such failure scenarios.

[1]Note that the AP and BS notions are interchangeably used in a 5G context

TABLE I
Notifications

Symbol	Description
$\mathcal{K}$	the universe of available videos in the video repository
$\mathcal{T}$	time frames set of the system
$I_k(t)$	the proportion of UAVs in *Initial state* at t
$P_k(t)$	the proportion of UAVs in *Pending state* at t
$C_k(t)$	the proportion of UAVs in *Caching state* at t
$T_k(t)$	the proportion of UAVs in *Transmission state* at t
$O_k(t)$	the proportion of UAVs in *Offline state* at t
$f(r)$	the probability density function of m located at r
p_a	the probability of m moving into area a
$\varphi_k(t)$	the proportion of UAVs that can cache content k at t
β_k	the requesting rate of content k
$\overline{E}_{k,t}$	the average request delay at t
$\overline{D}_{k,t}$	the average delivery delay at t
v_k	the eviction probability of cached content k
τ_k	the upper bound on the time interval
$E(\mathcal{T}_k)$	the average caching lifetime
$F_{l,a}$	the probability distribution of m moving out area a
N_a	the set of UAVs
$\mathcal{A}$	the set of area a
C_m	available caching space of UAV m

In the MDC^2 framework, the controller at each AP is mainly responsible for the following tasks. First, the controlling module collects the UAV status information and assigns each UAV a specific state to indicate its current actions related to a specific content piece, such as *caching* or *pending*. Secondly, the controlling module maintains the client status with the State Configuration Manager, as client status changes dynamically. Additionally, the Caching Estimator module that focuses on caching takes note of the changes in supply and demand. After determining the necessary copies of each content, the scheduling module in charge of the content distributes the caching responsibilities to the UAVs in their respective regions based on the recommended caching distribution method.

On the UAV side, the Caching Coordinator decides the allocation of cache content based on the request received by the AP. In order to achieve UAV-assisted video content distribution, the AP discovers U2U pairs according to the 5G-U2U profile and performs one-hop media content distribution, AP discovers U2U pairs based on the 5G-U2U profile for one-hop media content delivery. To be more precise, the AP first searches the network information database for a received request to identify a valid copy of the desired content. If UAVs containing the desired content are within range of the requester's U2U communication, content delivery can be done by creating a U2U link between the provider and the requester. Otherwise, the AP will process the request for content by establishing communication with the requester.

B. State Configuration Manager

This section presents the state space management and the proposed fluid-based model that characterizes how the population of UAVs in different states varies with the request arrivals, caching allocations, and data delivery. The notations used in the model are given in Table I.

1) State Space Management: We partition the network area into multiple non-overlapping areas, i.e., $\mathcal{A} = \{1, 2, \ldots, A\}$, whereby a given area $a \in \mathcal{A}$ is a circular region that is associated with the AP located at the center of the area a. The AP

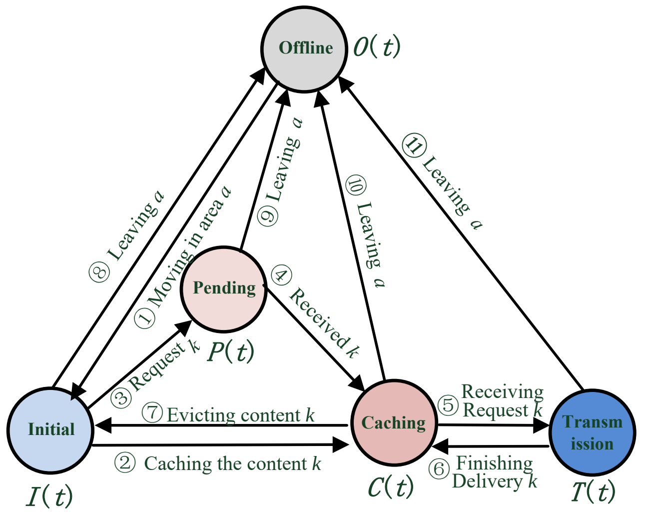


Fig. 2. State transition for MDC framework with novel *Transmission State* which represents UAVs that are delivering data to requesters.

receives data from a remote server and interconnects the UAVs to the controller in $a \in \mathcal{A}$. Let N_a denote the set of UAVs forming the MDC. Let $\mathcal{K}$ denote the universe of available videos in the video repository (remote server). The system time is divided into time frames, i.e., $t \in \mathcal{T} = \{1, 2, \ldots\}$. The UAV movements are assumed to follow the random waypoint model [22]. For a given time frame t, each UAV has one of the following three roles according to the operations related to distributing the video item $k \in \mathcal{K}$: *normal node* and it does not participate in the distribution of k; *consumer* as it is the requester of the content k; or *caching node* as it holds the content k in its caching space.

Based on the role of UAVs at different times, the following five states are defined:

1) *Initial state*: when an UAV moves into area a, it becomes a normal UAV in the initial state. We use $I_k(t)$ to represent the proportion of UAVs (with respect to the total number of UAVs in the considered area a) in the current state in a time frame t.
2) *Pending state*: when an UAV requests content k, it is in pending state; we use $P_k(t)$ to represent the overall proportion of UAVs in pending state.
3) *Caching state*: UAV enters the caching state when it has a copy of k and can provide it to other UAVs. The proportion of UAVs in this state at time t is denoted by $C_k(t)$.
4) *Transmission state*: an UAV in this state is delivering data to requesters. We denote $T_k(t)$ the population proportion in this state.
5) *Offline state*: the UAVs in this state have left the area a in the given time frame t; the proportion of UAVs in this state is represented by $O_k(t)$.

Note that for video k in a time frame t, the state of an UAV is unique, and the sum of the proportion of UAVs in each state is always one.

Transitions between these five states for any video item k are interpreted as shown in Fig.2. For each $k \in \mathcal{K}$, we describe the transition rates from one state to another as follows.

Transition 1: If an offline UAV moves into the area a, its state is set to *Initial*. According to previous analysis for circular areas in [23], the probability of any UAV m moving into area a can be given by the curve integral $p_a = \oint_{R_a} f(r)ds$, where R_a is a circular function with centre c (c indicates the AP coordinates, i.e., $c = (x_a, y_a)$) and radius R (communication range of AP of a). $f(r)$ is the probability density function of an arbitrary UAV m located at position r in Random Way Point (RWP) motion model [22]. Thus, the rate of Transition 1 is $p_a O_k(t)$.

Transition 2: If an UAV caches video $k \in \mathcal{K}$, its state is set to *Caching*. The caching parameter $\varphi_k(t)$ determines the proportion of UAVs that cache content k at time t. This parameter represents the caching scheme. For example, $\varphi_k(t) = 0$ indicates that none of UAVs in state $I_k(t)$ will cache item k.

Transition 3: If an UAV requests content k, its state is set to *Pending*. The arrival process of the request is further modeled as the Poisson process, and $\lambda_k \Delta t$ is used to represent the probability of content k being requested at time slot Δt, where λ_k is Poisson rate. When Δt is small enough, $\Delta t \to \mathrm{d}t$. Thus, we use $\beta_k I(t)$ to represent the rate of Transition 3, where β_k $=\lambda_k \Delta t$.

Transition 4: When an UAV finishes downloading the requested content k, its state changes to *Caching*. In our system, when the controller receives the request from UAV m, it first allocates a provider (i.e., an UAV in *Caching* state) to m. The probability of successfully allocating a provider for UAV m equals $C_k(t)/P_k(t)$. After the provider is selected, an U2U route between the provider and requester will be set up for data delivery. The time interval between the instant of the UAV request for a video item k and the instant of receipt of all required data depends on two types of delay: (1) Request delay: the delay between UAV requesting the video and its reception of the first video data packet. (2) Delivery delay: the time interval between receiving the first and the last video data packet. Let the average request delay and delivery delay in a given time frame t be denoted as $\overline{E}_{k,t}$ and $\overline{D}_{k,t}$. Thus, the rate of Transition 4 can be expressed as $C_k(t)/(\overline{E}_{k,t} + \overline{D}_{k,t})$.

Transition 5: When the AP receives a request for video k from an UAV in *Pending* state, the AP's Data Delivery Controller calculates the best provider for video k. The calculation is based on finding the UAV provider with the lowest total transmission delay. An UAV in *Pending* state may the provide video k to multiple other UAVs. If an UAV in state *Caching* $C_k(t)$ receives a request for content k, it enters the *Transmission* state $T_k(t)$. The corresponding probability can be given by the ratio between the population of pending UAVs and the population of UAVs holding k, i.e., $P_k(t)/C_k(t)$. Therefore, we use $P_k(t)$ to represent the rate of Transition 5.

Transition 6: After finishing the video transmission of content k, the UAVs in the *Transmission* state will enter state *Caching* $C_k(t)$. Similar to the Transition 4 case, the rate of Transition 6 can be expressed as $T_k(t)/(\overline{E}_{k,t} + \overline{D}_{k,t})$.

Transition 7: When the caching space for caching at an UAV is full, a caching content update is required. The eviction probability of cached content k is denoted by v_k. Based on the assumption that LRU is used, v_k can be calculated by [23]. Following [23], v_k is given by the inverse of the average caching lifetime, i.e., $1/E(\mathcal{T}_k)$, which can be derived as:

$$\begin{aligned} E(\mathcal{T}_k) &= E[t] - E[t_0] \\ &= \beta_k^{-1} e^{\beta_k} e^{\tau_k} - \frac{e^{-\beta_k \tau_k}(\tau_k + \frac{1}{\beta_k})}{e^{\beta_k \tau_k}} + \tau_k \end{aligned} \tag{1}$$

where τ_k denotes the upper bound on the time interval between consecutive caching hits. For example, if two consecutive requests for k are more than τ_k spaced apart, then item k will be evicted from the cache.

Transitions 8, 9, 10, and 11: As any UAV can leave the area, all states can become offline with a certain probability. Following [23], the probability distribution of any UAV moving out of the AP's communication range can be expressed by:

$$F_{l,a} = \int_h^k \frac{1}{b-a} \oint_{R_a} f(\delta_n | r) f(r) \mathrm{d}x\mathrm{d}y \tag{2}$$

where $f(\delta_n|r)$ represents the conditional probability that the UAV is located in area δ_n.

2) Fluid-based Modeling: The dynamic transitions of states can be characterized by the following ordinary differential equation with initial state $(I(t_0), P(t_0), C(t_0), T(t_0), O(t_0))$:

$$\frac{\mathrm{d}I_k(t)}{\mathrm{d}t} = f(r)O_k(t) - I_k(t)T_k(t) + v_k C_k(t) \tag{3}$$

$$\frac{\mathrm{d}P_k(t)}{\mathrm{d}t} = [\beta_k - F_{l,a}]I_k(t) - W_k(t)P_k(t) \tag{4}$$

$$\frac{\mathrm{d}C_k(t)}{\mathrm{d}t} = W_k(t)P_k(t) + \varphi_k(t)I_k(t) - F_{l,a}C_k(t) \tag{5}$$

$$\frac{\mathrm{d}T_k(t)}{\mathrm{d}t} = \varphi_k(t)I_k(t) - L_k(t)T_k(t) \tag{6}$$

$$\frac{\mathrm{d}O_k(t)}{\mathrm{d}t} = F_{l,a} - O_k(t)[f(r) + F_{l,a}] \tag{7}$$

with $T_k(t) = \beta_k + \varphi_k(t) + F_{l,a}$, $W_k(t) = T_k(t)/(\overline{E}_{k,T} + \overline{D}_{k,t}) - 1$, and $L_k(t) = \beta_k + E^{-1}(\mathcal{T}_k) + F_{l,a} + C_k(t)/(\overline{E}_{k,T} + \overline{D}_{k,t})$.

To represent the state in a time frame t, each UAV maintains a one-hot encoding $(\mathbb{I}_k(t), \mathbb{P}_k(t), \mathbb{C}_k(t), \mathbb{T}_k(t))$, whereby $\mathbb{I}$, $\mathbb{P}$, $\mathbb{C}$, $\mathbb{T}$ represent the *Initial*, *Pending*, *Caching*, *Transmission* state, respectively. Specifically, by collecting one-hot encoding from all UAVs, the AP can calculate the number of UAVs in the *Initial*, *Pending*, *Caching*, *Transmission* states for different media at a time frame t. The number of UAVs in the region shared between APs so that UAVs in the *Offline* state can be estimated. Further, the ratio of pending UAVs and the total number of UAVs can be obtained, denoted as the requesting rate β_k for each content. UAVs also report the moving speed and current location at each time slot to the controller for estimating the moving in/out rate.

IV. Caching Configuration

In this section, we discuss the function of caching coordinator in MDC^2 framework. The caching coordinator model is composed of the network caching estimator model and the UAV caching allocator model. In each time frame T, the network caching estimator model determines the optimal caching parameter φ_k of content k by collecting the initial state of the state configuration manager model, and the UAV caching allocator model describes the process of caching allocation according to φ_k. In the following part, we will explain the detailed working process of network caching estimator model and UAV caching allocator model by describing the making of caching decision and the design of caching algorithm.

A. Caching Decision Making

In an UAV-assisted video content distribution scenario, to determine the optimal caching parameter φ_k, it is necessary to consider the limited caching resources of UAVs and the associated communication latency and cost. On one hand, excessive video caching can reduce both communication costs and access latency, but it consumes a lot of caching resources. On the other hand, although reducing the video cache ratio can reduce the consumption of caching resources, the number of video requests increases correspondingly, and the communication cost and video access delay between UAVs become larger, reducing the quality of service. Therefore, the optimal caching allocation strategy should balance the number of requests and caching consumption. The number of video requests at a time frame T is positively correlated with the proportion of UAVs in *Pending* state, i.e., $P_k(t)$. Assuming that UAVs contribute with the same amount of caching space when making caching decisions, then caching consumption is positively correlated with the proportion of UAVs in *Caching* state, i.e., $C_k(t)$. So caching decision's objective function in each frame T can be formulated as:

$$J_T(\varphi_k) = \alpha P_k(T^*) + \beta C_k(T^*) \tag{8}$$

where $\alpha + \beta = 1$ and $T^* = \arg\max_{t \in T} P_k(t)$. T^* is the moment when the maximum number of requests occurs in time frame T, which is used to represent the peak load of the system. $J_{T,a}(\varphi_k)$ can be derived by solving equations (3)–(7) via numerical methods, i.e., the Runge-Kunta method. The goal is to optimize the overall caching configuration over the entire time horizon $\mathcal{T}$. Therefore, the caching optimization problem is defined as follows:

$$\text{Minimize} \quad \sum_{a=1}^{N} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} J_{t,a}(\varphi_k) \tag{9}$$

$$\text{S.t} \quad 0 \le \varphi_k \le 1 \tag{10}$$

In our system, controllers share UAV count information to estimate the number of offline UAVs. However, even though this information is shared among controllers, each controller makes decisions based solely on its initial state, independent of the states of other controllers. Based on the formulated problem expressed in equations (9) and (10), the decision making process of each controller is separable. $\sum_{a=1}^{N}$ from equation (9), means that the optimum of (9) can be solved individually for each objective $\sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} J_{t,a}(\varphi_k)$ at each controller a. From the formulation perspective, the problem can be distributed to each controller as follows:

$$\text{Minimize} \quad \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{T}} J_{t,a}(\varphi_k) \tag{11}$$

$$\text{S.t} \quad 0 \le \varphi_k \le 1 \tag{12}$$

In each time slot t, we first calculate $J(\varphi_k)$ for each of k by introducing all $\varphi_k \in [0, 1]$ into equations (3)–(7), and selecting the optimal $\varphi_{k,T}$ by:

$$\varphi^*_{k,t} = \arg\min_{\varphi_k\in(12)} J_{t,a}(\varphi_k) \tag{13}$$

Throughout, we assume that a time frame t is long enough for caching operations. At the beginning of a time frame, the controller estimates the caching demand based on the fluid-based model, and then it makes caching decision for optimizing the content distribution in the current time frame.

B. Caching Algorithm Design

We propose two different caching algorithms to illustrate the process of caching allocation: (1) probabilistic caching; and (2) stable-preferred caching.

Probabilistic Caching: We define C_m as the available caching space of UAV m. In each T, the AP broadcasts caching replicas for each content sequentially in time frame T by the descending order of φ^*_k, the controller sorts the UAVs in the descending order of C_m. For each UAV in IoT system, if the UAV has free caching, then it will cache the content k with probability φ^*_k. If the UAV's caching is full and it is determined to cache k, the UAV will free up caching space by evicting some content according to the LRU replacement rule. The pseudo-code for the above iteration is shown in Algorithm 1.

Algorithm 1: Probabilistic caching algorithm in time frame T

Input: C_m: available caching space of UAV m, requesting probability distribution
Output: Caching configuration

for $k \in \mathcal{K}$ **do**
  $\varphi^*_k = \arg\min_{\varphi_k\in(12)} J_{T,a}(\varphi_k)$;
end
Sort UAVs in area a in descending order of C_m;
Sort videos in $\mathcal{K}$ in descending order of φ^*_k;
Controller side:
while $k \in \mathcal{K}$ **do**
  Broadcast k;
end
UAV m's **side**:
while $k \in \mathcal{K}$ **do**
  Determine whether to store k with φ_k;
  if k *is determined to cache* **then**
    if $C_m > 0$ **then**
      Cache k;
      $C_m \leftarrow C_m - 1$;
    end
    else
      Proceed LRU;
      Cache k;
    end
  end
end

Stable-preferred Caching: The content with high caching probability will be preferably placed at UAVs that have high probability of staying in the current area a (i.e., are not likely to leave the area). This is important because the number of replicas of the content will change when an UAV moves in and out of an area. When an uav moves out of an area, the content cached on it cannot be requested, resulting in a situation where the demand for these contents is higher than their availability (i.e. a cache shortage). In this case, the request delay for these contents will increase. Besides, when the UAV moves out of its area, the number of replicas cached by the mobile UAVs in the area needs to be increased to deal with the cache shortage problem. In the case of UAVs entering an area, multimedia service performance will not be affected, but content replicas will still need to be redistributed. Therefore, in order to diminish the influence of UAV's movement behavior, we propose the stable-preferred caching algorithm that puts content into UAVs with high stability. We define $S(m)$ as the stability of UAV m. To differentiate the stability level of UAVs, we jointly consider the average moving speed $\overline{v}_m$ and its distance D_m to the edge along the direction of speed:

$$S(m) = D_m/\overline{v}_m \tag{14}$$

A large value of $S(m)$ means that the UAV m has a high probability of staying in the current area for a period of time in the future, and thus has a high stability. We also consider content popularity, where popularity is a network-level concept independent of area. Unlike (1) in which each replicas of content is cached in descending order of φ^*_k, in (2) we consider that content with higher popularity should be cached preferentially. Content popularity represents the arrival rate of content requirements in the future period of time. Caching each content in descending order of popularity will increase the caching hit rate. In MDC^2 framework, we represent the popularity of the content k by the proportion of UAVs in caching state at t, i.e., $C_k(t)$. The regular update of $C_k(t)$ reflects the changing demand for content k in real time. To sum up, in Stable-preferred Caching algorithm, during each frame T, the controller sorts the contents and UAVs in the descending order of $C_k(t)$ and $S(m)$, respectively. $C_k(t)$ represents the popularity of the content, and its value will be updated periodically. Let the number of UAVs in the area a be denoted as $N_a(T)$. For the each k, the first $\lfloor \varphi^*_k N_a(T) \rfloor$ will be outputted from N_a and selected to cache the k. The pseudo-code for the above iteration is shown in Algorithm 2.

V. Mobility-aware Enhanced Fast U2U Video Delivery

In this section, we formulate the U2U video delivery problem and then solve it by proposing a smooth playback-ensured fast delivery policy (SPFDP) for selecting the content providers with the lowest delay while also smoothening the video playback.

Algorithm 2: Stable-preferred caching algorithm in time frame T

Input: $N_a(T)$: the number of UAVs in the area a, N_a: a set of UAVs, $S(m)$: the stability of UAV m, C_m: available caching space of UAV m, $|\mathcal{N}|$: the minimum number of UAVs to cache video k, requesting probability distribution, $C_k(t)$

Output: Caching configuration

for $k \in \mathcal{K}$ **do**
  $\varphi_k^* = \arg \min_{\varphi_k \in (12)} J_{T,a}(\varphi_k)$;
end
Sort UAVs in N_a in descending order of $S(m)$;
Sort videos in $\mathcal{K}$ by the descending order of $C_k(t)$;
Controller side:
while $k \in \mathcal{K}$ **do**
  if $\lfloor \varphi_k^* N_a(T) \rfloor \geq |\mathcal{N}|$ **then**
    Select the first $\lfloor \varphi_k^* N_a(T) \rfloor$ UAVs to cache k;
  end
  else
    Select $|\mathcal{N}|$ UAVs to cache k;
  end
end
UAV m's **side**:
while $k \in \mathcal{K}$ **do**
  if *m is selected to cache k* **then**
    if $C_m \geq 0$ **then**
      Cache k;
      $C_m \leftarrow C_m - 1$;
    end
    else
      Proceed LRU;
      Cache k;
    end
  end
end

As described in the fluid-based model described in equations (3)–(4), the request and delivery delays affect the efficiency of the requester and the busy nodes. Lower delays not only improve the QoS, but also the cache utilization. Assume an IoT device on the ground sends a request for video k, which is accepted by UAV n and forwarded to the controller of the corresponding AP. When the controller receives the request for video k from UAV n, it will select a provider for delivering data k. Let P_k denote the set of providers of video k. For each $m \in P_k$, let $d(m,n)$ and $B(m,n)$ represent the current end-to-end delay and bandwidth between m and n, respectively. Let V_k and B_k denote the size and bitrate of k, respectively. We then formulate the problem of selecting provider for n as follows:

$$\min D_m = d(m,n) + V_k/B(m,n) \tag{15}$$
$$B(m,n) \geq B_k \tag{16}$$
$$m \in P_k \tag{17}$$

The objective of equation (15) is to minimize the total delay, and the constraint from equation (16) enforces that the bandwidth is larger than the bitrate of k, thereby ensuring smooth playback. The equation (17) constraint ensures that the candidate provider m holds the cached replica of item k.

By solving the above problem, we design SPFDP. For a request from an UAV n, the controller rearranges the candidate providers in P_k in ascending order of their D_m, i.e, for $m, j \in P_k$, $D_m \geq D_j$ for $m < j$. This can be achieved by using, for instance, the quicksort algorithm with a complexity of $O(N \log N)$. The optimal solution is:

$$m^* = \arg \min_{m \in P_k, B(m,n) \geq B_k} D_m \tag{18}$$

As a result, the controller derives m^* with Algorithm 3.

Based on the SPFDP, we present the overall design of our U2U-based video delivery scheme:

(1) UAV n receives a request for k from the IoT device and then issues the request for item k to the AP of its residential area a. The AP forwards the request to the controller for processing.
(2) If P_k at the controller of area a is non-empty (P_k denotes the set of UAVs carrying the content k in area a), then the controller will execute Algorithm 3 to select UAV m as provider and establish the U2U route from provider UAV m to the requesting UAV n.
(3) If P_k is empty, the controller of area a will inquire with the controllers of its neighbourhood areas about requesting item k. If there exists a neighbor area b with non-empty P_k, then the controller of area b executes Algorithm 3 with the controller of area a as input, extracts the data from its provider UAV m and forwards it to the controller of a via the fronthaul link between the APs in a and b. Then, the controller of area a forwards item k to the requesting UAV n.
(4) If item k is not stored in either a or its neighbours, then the request will be directly forwarded to the video server and the server will return the video content to n via the AP. The AP acts as a forwarder only.
(5) UAV n forwards k to the IoT device on the ground via UAV-to-Device (U2D) communications.

VI. Performance Evaluation

A. Set-up

The simulations are performed with Python on a computer with an Intel i7-12700 CPU and 32 GB RAM. We consider a network area of 3000×3000 m^2 with 12 5G-NR APs deployed at arbitrary locations within the network area. Each AP has a communication range of 500 m. The number of total UAVs is set to 300, and each UAV is capable of 5G-U2U communication. The U2U link has a communication range of 150 meters and a bandwidth of 30 Mbps. 20 media instances are used in the experiment, each of which has a duration of 200 seconds. The video segments have a length of 2 seconds and a bitrate of 8 Mbps. The arrival of video requests for each instance follows a Poisson distribution. The parameter λ of the Poisson distribution is randomly selected between 2 and 20.

Algorithm 3: Provider selection for the request for content k

Input: Requester n, P_k: the set of UAVs carrying the content k, $d(m,n)$: end-to-end delay between m and n, $B(m,n)$: bandwidth between m and n, V_k: the size of content k, B_k: the bitrate of content k
Output: Provider m^*

for $m \in P_k$ **do**
  $D_m = d(m,n) + V_k / B(m,n)$;
end
Sort P_k in ascending order of D_m;
$m = 1$;
while $m \in P_k$ **do**
  if $B(m,n) \geq B_k$ **then**
    Return m;
  end
  else
    $m++$;
  end
end

The simulation time is 1000s and 95% confidence intervals are evaluated. We assume that the motion model of UAV is RWP. Given the moving velocity range, each UAV can independently change its own position during the simulation.

The following two performance metrics are considered:

- Average access latency (AAL): After sending the request, the waiting time to receive the first packet can be defined as access latency. The average of access latency for all UAVs is defined as AAL.
- Caching hit ratio (CHR): CHR is defined as the ratio at which video requests from UAVs can be satisfied.

The advanced Random-Cache algorithm introduced in [23] will be compared with the MDC^2 algorithm.

B. Simulation Results

1) Impact of Caching Size: Caching size refers to the total amount of content an UAV can store. Caching size determines how many mobile devices an UAV can serve and how many different kinds of content it can cache. In UAV-based IoT networks, the size of the caching directly affects the performance of the edge-assisted caching. Figures 3 plot the AAL and Figures 4 plot the CHR results for different caching sizes when Random, MDC2-Sta, and MDC2-Pro are used in sequence. MDC^2-Sta and MDC^2-Pro stand for stable-preferred caching and probabilistic caching, respectively, as described previously.

Each data point in Fig. 3 indicates the AAL at the end of the simulation, and the data point in Fig. 4 represents the average CHR during the simulation, with varying caching space from 1% to 4% of the maximum UAV caching space. According to Fig. 3, with the increment of caching size, the AAL of each algorithm has a decreasing trend. This phenomenon can be easily understood since the likelihood of accessing content within a single hop is directly proportional to the caching capacity. However, MDC^2-Sta outperforms MDC^2-Pro and the random strategy [23] with up to 11% and 17% lower AAL. In Fig. 4, the CHR of three methods increases with the increment of total caching size. This can be attributed to the fact that a higher caching hit rate can be achieved with a larger caching size yield. The CHR of both MDCMDC^2 methods is higher than that of the Random Cache [23]. This can be attributed to the precise and prompt estimations of content demands. For example, with 4% of the cache size, MDC^2-Pro achieves a higher CHR than random caching. Compared with MDC^2-Pro, MDC^2-Sta achieves higher CHR, i.e., 20% higher than MDC^2-Pro at size 4%. The reason for this improvement is mainly because content with higher popularity is placed on more stable devices, which increases the caching hit rate.

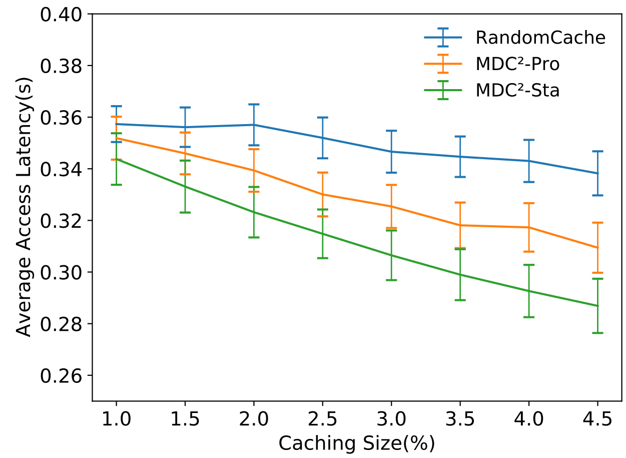


Fig. 3. Average Access Latency between different cache size

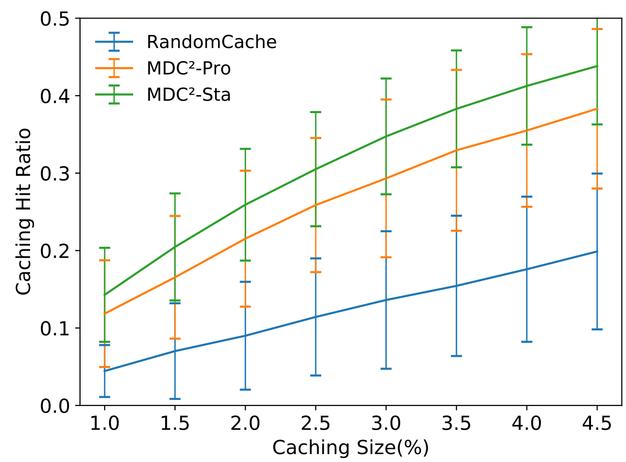


Fig. 4. Caching Hit Ratio between different cache size

2) Evolution over Simulation Time: Simulation time is the length of time for evaluating the performance of the caching framework. The longer the simulation time is, the more behaviors and results of the caching framework can be collected and observed, and the more reliable the results analysis is. Simulation time is an important factor in evaluating the performance of caching frameworks. By increasing the simulation time, we can accurately evaluate the performance of

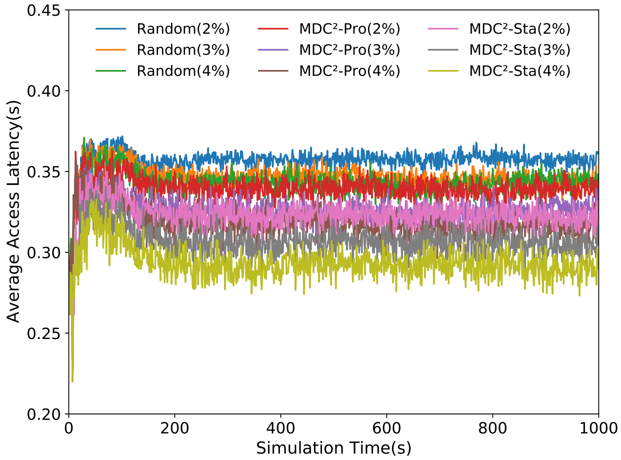


Fig. 5. Average Access Latency with simulation evolution

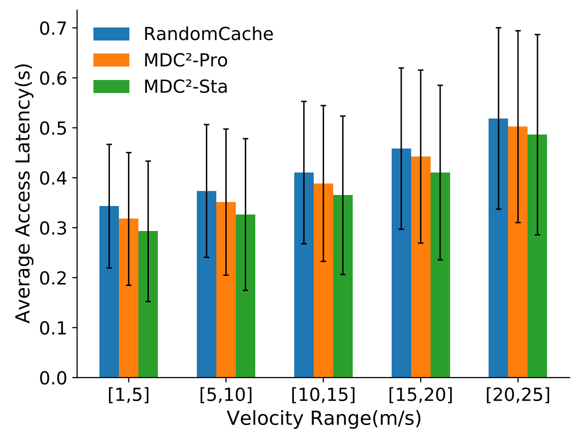


Fig. 7. Average Access Latency between different velocity

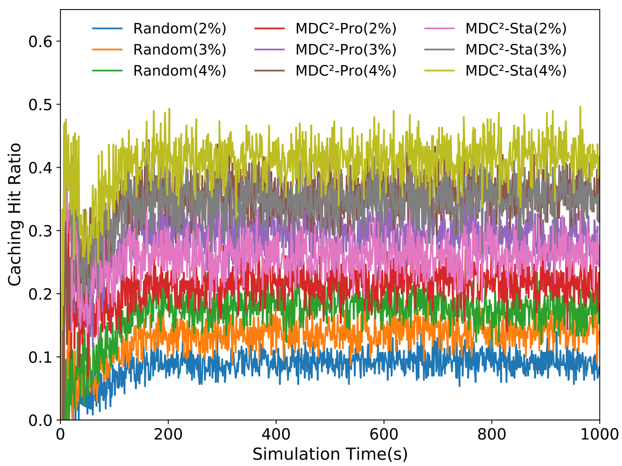


Fig. 6. Caching Hit Ratio with simulation evolution

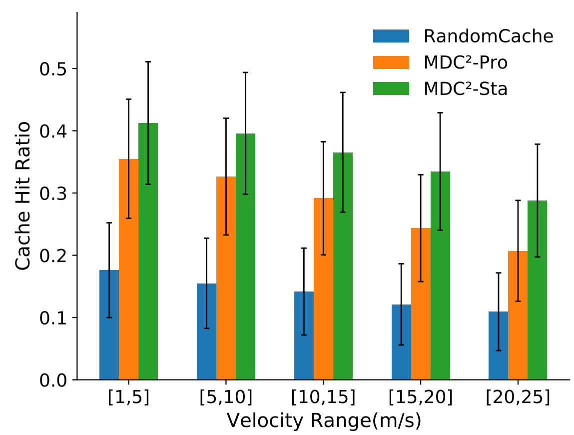


Fig. 8. Caching Hit Ratio between different velocity

the caching framework over long runs. Figs. 5 and 6 represent the relationship between AAL/CHR and simulation time. Fig. 5 shows the increasing trend of AAL value in the early stage of simulation. The increment of curves is due to the increasing number of requests. All AAL values increase at the start-up stage and gradually become stable. The overall MDC^2 AAL is lower than that for random caching [23]. Take the caching capacity of 4% as an example, the AAL curve of MDC^2-Pro is lower than that of random caching, and the AAL of MDC^2-Sta is even lower.

Fig. 6 shows how CHR fluctuates much during the simulation. In general, the CHR value of MDC^2 is higher than Random Cache due to its dynamic allocation of cache space for each content based on future demand estimates. Among the two MDC^2 solutions, MDC^2-Sta outperforms MDC^2-Pro; for example, for caching size 4%, MDC^2-Sta has lower CHR than MDC^2-Pro.

3) Variation of Moving Velocity: UAV Velocity is the speed at which an UAV travels through the air. Changes in UAV velocity affect communication between UAVs and the proportion of UAVs in different states, resulting in dynamic changes in content requests and caching allocations. Figs. 7 and 8 present the results in terms of AAL and CHR, for different UAVs' moving speeds, when the two versions of MDC^2 and Random Caching are used in turn. Five velocity intervals are considered, which start from 1m/s and up to 25m/s. As shown in Fig. 7, AAL increases with the increment of UAV moving speed. With higher velocity, U2U link fluctuations increase drastically and further increase AAL. MDC^2-Sta has the lowest AAL and the Random Caching scheme has the highest AAL. As shown in Fig. 8, the decreasing trend of CHR can be observed with the increment UAV velocity.

Notable is that in general, MDC^2-Sta outperforms the other two solutions in terms of AAL and CHR for each velocity range. For instance, if the moving speed is between 20 and 25 m/s, MDC^2-Pro's CHR and AAL are 25% higher and 6% lower than MDC^2-Sta, respectively, and more than 100% and 11% higher than those obtained by Random Cache [23].

VII. Conclusions and Future Work

This work proposed a novel MDC^2 framework to address the caching allocation problem for multimedia streaming in

IoT networks composed of UAVs. In this framework, we formulated a fluid-based model to precisely describe the dynamic behavior of each UAV and estimate the resource demand for different video content. Additionally, an online optimization algorithm for caching allocation problem was designed, based on the fluid-based model. Extensive experiments were performed and the results showed that MDC^2 can significantly enhance the average Caching Hit Ratio (typically by 40%) and decrease the Average Access Latency (typically by 25%) when compared with the current state-of-the-art algorithm: Random-Cache algorithm. Future work will focus on the energy consumption of UAV in IoT networks and QoS issues in time sensitive networks.

VIII. Acknowledgement

This work is supported by the National Key R&D Program of China under grant no. 2023YFB2904400. It is also supported by the open research fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education under grant JZNY202302. This work is supported by the National Natural Science Foundation of China (No. 62225105, No. 62394323, No. 92167204, No. 62072030, No. 62001057); It is also supported by Zhejiang Lab Open Research Project (N0.K2022QA0AB05). Gabriel-Miro Muntean would like to acknowledge the support of the Science Foundation Ireland grants 21/FFP-P/10244 (FRADIS) and 12/RC/2289_P2 (INSIGHT).

References

[1] Cisco, *Cisco Annual Internet Report(2018–2023)*.

[2] C. Long, et al., "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE TMM*, vol. 20, no. 5, pp. 1126–1139, 2018.

[3] V. Balasubramanian, M. Wang, M. Reisslein, and C. Xu, "Edge-Boost: Enhancing multimedia delivery with mobile edge caching in 5G-D2D networks," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, July 2019, pp. 1684–1689.

[4] Yi-Han Xu, Meng-Lian Liu, Jing-Wei Xie, and Jun Zhou, "Media independent mobility management for d2d communications over heterogeneous networks (hetnets)," *Wireless Personal Communications*, vol. 120, no. 4, pp. 2693–2710, 2021.

[5] A. Boukerche, R. W. L. Coutinho, and A. A. F. Loureiro, "Information-centric cognitive radio networks for content distribution in smart cities," *IEEE Network*, vol. 33, no. 3, pp. 146–151, May 2019.

[6] Khaled F Hayajneh, Ferdos Abualkishik, Khaled Bani-Hani, Hazim Shakhatreh, and Zaid Albataineh, "Efficient three-dimensional deployment of multiple unmanned aerial vehicles supporting ground base station toward maximizing served users with heterogeneous quality-of-service requirements," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 9, pp. e4818, 2023.

[7] Mostafa M Abdelhakam, Mahmoud M Elmesalawy, Ibrahim I Ibrahim, and Samir G Sayed, "Two-timescale optimization approach for coordinated multi-point design in unmanned aerial vehicle-assisted cellular networks," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 1, pp. e4676, 2023.

[8] Pradip Kumar Barik and Raja Datta, "Svm-d: Efficient shvc video multicast through d2d offloading over 5g cellular networks," in *2022 National Conference on Communications (NCC)*, 2022, pp. 124–129.

[9] Zhuo Li, Wentao Dong, and Xin Chen, "Deep reinforcement learning-based incentive mechanism design for short video sharing through D2D communication," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 6, pp. 3946–3958, 2021.

[10] Chao Chen, Tiankui Zhang, Yuanwei Liu, Geoffrey Ye Li, and Zhimin Zeng, "Joint user association and caching placement for cache-enabled uavs in cellular networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 1–6.

[11] Erbao Wang, Qunfeng Dong, Yongan Li, and Yuye Zhang, "Content placement considering the temporal and spatial attributes of content popularity in cache-enabled uav networks," *IEEE Wireless Communications Letters*, vol. 11, no. 2, pp. 250–253, 2022.

[12] Bin Zeng, Cheng Zhan, Yang Yang, and Jingrui Liao, "Access delay minimization for scalable videos in cache-enabled multi-uav networks," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 389–394.

[13] Jingjing Luo, Jialun Song, Fu-Chun Zheng, Lin Gao, and Tong Wang, "User-centric uav deployment and content placement in cache-enabled multi-uav networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5656–5660, 2022.

[14] Qiang Fan and Nirwan Ansari, "Towards traffic load balancing in drone-assisted communications for iot," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3633–3640, 2019.

[15] Fahimeh Fazel, Jamshid Abouei, Muhammad Jaseemuddin, Alagan Anpalagan, and Konstantinos N. Plataniotis, "Secure throughput optimization for cache-enabled multi-uavs networks," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7783–7801, 2022.

[16] Congshan Fan, Xu Zhou, Tiankui Zhang, Wenqiang Yi, and Yuanwei Liu, "Cache-enabled uav emergency communication networks: Performance analysis with stochastic geometry," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 7, pp. 9308–9321, 2023.

[17] Wenqin Zhuang, Yiming Song, Mingkai Chen, Jianxin Chen, and Haibo Li, "Content sharing strategy for blind popularity distribution in d2d communications," *Transactions on Emerging Telecommunications Technologies*, p. e4830, 2023.

[18] Manoj Kumar Somesula, Sai Krishna Mothku, and Anusha Kotte, "Deep reinforcement learning mechanism for deadline-aware cache placement in device-to-device mobile edge networks," *Wireless Networks*, vol. 29, no. 2, pp. 569–588, 2023.

[19] Nannan Zhang, Wei Wang, Ruining Lan, Pan Zhou, and Aiping Huang, "Delay-aware cache-enabled cooperative d2d transmission in mobile cellular networks," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.

[20] Tongyu Zong, Chen Li, Yuanyuan Lei, Guangyu Li, Houwei Cao, and Yong Liu, "Cocktail edge caching: Ride dynamic trends of content popularity with ensemble learning," *IEEE/ACM Transactions on Networking*, pp. 1–12, 2022.

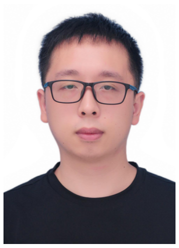
[21] Manoj Kumar Somesula, Rashmi Ranjan Rout, and D.V.L.N. Somayajulu, "Cooperative cache update using multi-agent recurrent deep reinforcement learning for mobile edge networks," *Computer Networks*, vol. 209, pp. 108876, 2022.

[22] David B Johnson and David A Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile computing*, pp. 153–181, 1996.

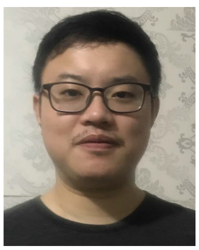
[23] C. Xu, M. Wang, X. Chen, L. Zhong, and L. A. Grieco, "Optimal information centric caching in 5G device-to-device communications," *IEEE Trans. Mob. Comp.*, vol. 17, no. 9, pp. 2114–2126, Sep. 2018.

Lujie Zhong received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2013. She is currently a Professor with the Information Engineering College, Capital Normal University, Beijing. She has published papers in prestigious international journals and conferences in the related area, including IEEE Comm. Magazine, IEEE TMC, IEEE TMM, IEEE IoTJ, IEEE INFOCOM, and ACM MM. Her research interests include communication networks, computer system and architecture, and mobile networks.

Shujie Yang received the Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2017. He is an Associate Professor with Beijing University of Posts and Telecommunications. His current research interests include the areas of VR networks, content delivery network, and wireless networking.



Kefei Song received his B.S. degree in Computer science and Technology from Northwestern Polytechnical University, Xi'an Shaanxi, China in 2021. He is currently working toward a master's degree in School of Computer Science, Beijing University of Posts and Telecommunications. His research interests include computer networks and reinforcement learning.



Mu Wang received the Ph.D. degree in computer technology from the Beijing University of Posts and Telecommunications (BUPT) in 2020. He currently serves as a Post-Doctoral Researcher with the Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University. His research interests include information centric networking, wireless communications, and multimedia sharing over wireless networks.

Ke Jiang received her B.S. degree in Telecommunications Engineering and Management from Beijing University of Posts and Telecommunications, Beijing, China in 2022. She is currently working toward a master's degree in School of Computer Science, BUPT. Her research interests include multimedia delivery and wireless networking.

Gabriel-Miro Muntean (Fellow, IEEE) is a Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and co-Director of DCU Performance Engineering Laboratory. He has published four books and over 500 papers in top international journals and conferences. His research interests include rich media delivery quality, performance, and energy-related issues, technology enhanced learning, and other data communications in heterogeneous networks. He is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, the Multimedia Communications Area Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and reviewer for important international journals, conferences, and funding agencies. He coordinated the EU project NEWTON and led the DCU team in the EU projects TRACTION and HEAT.