IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON INFORMATION CENTRIC WIRELESS NETWORKING WITH EDGE COMPUTING FOR 5G AND IOT

Received March 15, 2020, accepted April 20, 2020, date of publication April 22, 2020, date of current version May 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989532

Scalable Video Caching for Information Centric Wireless Networks

ZHILONG ZHANG[1], (Member, IEEE), JIANMEI DAI[1], (Member, IEEE), MINYIN ZENG[1], DANPU LIU[1], (Senior Member, IEEE), AND SHIWEN MAO[2], (Fellow, IEEE)

[1]Beijing Laboratory of Advanced Information Network, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, USA

Corresponding author: Zhang Zhilong (zhilong.zhang@outlook.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61801051and Grant 61971069, in part by the Beijing Natural Science Foundation under Grant L172032, and in part by the National Science Foundation under Grant IIP-1822055.

ABSTRACT Recently, information centric wireless network (ICWN) has been concerned due to its flexible network structure and high efficiency of content delivery. Meanwhile, scalable video coding (SVC) is a promising solution to provide high quality of video services. Combining ICWN and SVC is expected to improve the performance of wireless video delivery services. Since caching strategy plays a significant role in ICWN, in this paper, we address the caching problem for scalable videos over ICWN in mobile scenarios. By jointly considering the layered structure of SVC and the hierarchical architecture of ICWN, we formulate an optimization problem to minimize the average download delay. A novel layered hierarchical caching method is proposed for solving the problem. Furthermore, we focus on a special but common case in which the above delay minimization problem is equivalently transformed into a cache hit ratio maximization problem. A simplified algorithm with 1/2 approximation ratio is provided. Finally, simulation results show that our proposed caching schemes outperform baseline methods in cache hit rate and delay performance.

INDEX TERMS ICWN, SVC, wireless caching, video transmission.

I. INTRODUCTION

With the development of wireless communication technologies and the proliferation of smart devices, we have witnessed the explosive increment of mobile traffic. According to [1], the total amount of mobile traffic is predicted to reach 136 exabytes (EB) per month by the end of 2024, and over 74% of the traffic will be contributed by videos. Meanwhile, it have been observed that the majority of the traffic is generated from some replicate popular video contents [2].

To effectively cope with such a traffic growth, network operators aim at not only expanding the network capacity, but also changing the network structure to prevent unnecessary traffic forwarding and large delays. Information centric wireless networks (ICWN) is regarded as a promising technology to achieve the aforementioned goals. Different from the content delivery network (CDN) which caches contents in the application layer outside the access networks, ICWN provides in-network caching capabilities nearby mobile users, and reduces the average download delay when the requested contents are locally cached. Moreover, benefiting from edge computing, ICWN has the potential to learn users' behavior, such as mobility pattern and request distribution. Compared with traditional information centric networking (ICN), ICWN offers unique opportunities, since proactive caching can be possible in the wireless nodes before user demand.

In addition, with respect to adaptive video streaming services, the same content may be downloaded with various bit rates due to users' inhomogeneous screen resolutions and channel conditions. As a part of the H.264/265 standards, scalable video coding (SVC) can adjust to various network conditions and user requirements, while guaranteeing acceptable video quality. Specifically, SVC encodes a video into a base layer and multiple enhancement layers [3]. The base layer provides a fundamental video quality and carries essential information, and the enhancement layers provide different levels of improved video qualities. Before decoding of a higher enhancement layer, the base layer and all lower enhancement layers should be correctly decoded [4]. The caching of SVC-based contents can be more efficient if its layered video structure is considered.

Given that the scalability provided by SVC is beneficial for video delivery and ICWN has been widely concerned due to

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Seo Kim.

its flexible network structure, SVC-based video over ICWN is expected to improve the performance of wireless content delivery services. Therefore, in this paper, we consider such a video steaming scenario, and focus on caching strategies for better video delivery.

Caching in wireless networks has been widely concerned [5]–[10]. The concept of wireless caching is introduced in [5] to alleviate the explosive increase in video-on-demand transmissions. Popular video files are cached in equipments with high storage capacity to assist the macro base stations by handling local requests. After that, a large number of researchers study the issues of wireless caching in different aspects. For example, in [6], [7], the authors focus on the hierarchical structure of networks, and provide insights into the design of cooperative caching algorithms based on the topology of hierarchical tree. In [8], the authors aim at minimizing the average delay-cost of content delivery in cloud radio access networks through cooperative hierarchical caching. The work in [9] extends the terrestrial wireless caching into the sky, and addresses the problem of proactive content deployment in cache-enabled unmanned aerial vehicles. The authors in [10] consider dynamic video streaming in device-to-device assisted wireless networks, and design a method to cache video files of varying quality levels to enhance the quality of user experience (QoE). However, all the above mentioned researches take no consideration of caching SVC-based contents, and thus cannot fully exploit the benefits of layered videos.

To utilize the scalability provided by SVC, wireless caching dedicated for SVC-based contents have also been actively studied [11]–[18]. The authors in [11] investigate cache-enabled wireless networks to provide scalable video services with multiple perceptual qualities. Based on the theory of stochastic geometry, the expressions of local serving probability, ergodic service rate and service delay are derived. To improve energy efficiency (EE), the authors in [12] propose energy-efficient caching schemes for SVC-based videos over heterogeneous networks. In [13], the problem of joint power allocation and SVC-based content caching is addressed, and the QoE is improved by emphasising user's reception capability. For improving successful transmission probability, the authors in [14] investigate a layer placement scheme for SVC videos, where multiple video layers are stored in the cache devices of small-cell base stations (SBSs). In [15], caching strategies in large-scale wireless networks are analyzed and optimized, which reveals the relationship between layers of SVC-based videos. Moreover, some recent studies address the transmission latency in SVC-based video caching. The policies proposed in [16], [17] aim to reduce the average delivery delay of SVC videos in content delivery networks and heterogeneous wireless network. In [18], the authors provide an analytical characterization of the video delivery delay in a cache-enabled network, where the available video layers are stored based on their popularity. However, all of the above SVC-based caching schemes have not considered user mobility in wireless networks, and may not be applied with high efficiency when users are moving from one access point to another.

In addition, caching strategies play a significant role in traditional ICNs or content centric networks (CCNs), where routers are cache-enabled. There are already some researches on the caching issue for SVC-based videos over ICN. For instance, the authors in [19] propose a mechanism for cache management and request forwarding policies for scalable video streaming in ICN, which provides video faster to the users, especially the mandatory base layer. In [20], to improve the QoE for adaptive scalable video streaming services, layered cooperative cache management (LCC-VCCN) scheme is proposed. Neighbor nodes within broadcast range are selected to cache one or several SVC layers content, which reduces content retrieval time and prevents stalls of the video playback. However, the wireless factors such as channel fading, cell association, and user mobility involved in ICWN have not been jointly considered in the aforementioned caching strategies.

Motivated by the above discussion, we would like to design proactive caching schemes for SVC-based videos over ICWN, and take into account the following specific characteristics. Firstly, caching more layers results in higher scalability for video delivery. However, the cost will increase accordingly. Therefore, the number of video layers should be cached properly. Secondly, a mobile user requesting adaptive video streaming services often selects an appropriate video bitrate according to its channel bandwidth [21]. The factors which have important impact on the download rate should be taken into consideration, such as channel fading, cell association, and user mobility. Thirdly, the hierarchical network architecture of ICWN and layered structure of SVC videos have to be jointly considered. In summary, the novelty and technical contributions of this work are as follows:

- We address the problem of scalable video caching over ICWN, and aim to minimize the average transmission delay. Both the hierarchical caching architecture of ICWN and the layered feature of SVC-based videos are taken into account. An optimization problem is formulated and proved to be NP-hard. By simplifying it into a special knapsack problem and solving it through machine learning, we propose a layered hierarchical caching scheme.
- We consider a special case of the above delay minimization problem, which can be equivalently transformed into a simple cache hit rate maximization problem. A heuristic algorithm is proposed which provides at least $1/2$ approximate ratio of the optimal solution.
- Simulation results show that our proposed caching strategies achieve improvements in both transmission latency and cache hit rate compared to baseline caching strategies.

The rest of the paper is organized as follows. In Section II, we present the system model, and formulate a delay minimization problem for wireless SVC-based videos over

TABLE 1. Symbol notation.

Symbols	Descriptions
$\mathcal{K}$	Set of APs
$\mathcal{V}$	Set of videos in library
$\mathcal{L}$	Set of video versions or layers
$\mathcal{U}$	Set of all mobile users
$\mathcal{U}_k$	Set of users that will be associated with AP k
$\ell_u(t), z_u(t)$	Location and velocity of user u at time t
r_{ku}	Distance between user u and AP k
λ_{kv}^{um}	Probability of user $u \in \mathcal{U}_k$ to request segment (v, m)
o_{vl}	Average size of the l-th layer of video segment (v, m)
T_v	Segment duration of video v
C_0, C_k	Capacity of the caches at edge controller, AP k
d_R, d_0, d_k	Average single bit delay of downloading videos from Internet, edge controller, AP k.
$\mathcal{S}_0, \mathcal{S}_k$	Set of video items cached by edge controller, AP k
b_{0vl}^m, b_{kvl}^m	Indicate whether the l-th layer of video segment (v, m) is cached by edge controller, AP k
p_{kvl}^u	Probability that current transmit rate of user u in AP k can afford the streaming of the lth version of video v
q_{kvl}	Total uncached data of l layers of video v in AP k
α	The parameter to shape a Zipf distribution
γ_{ku}	Signal-to-noise ratio of the received signal at UE u

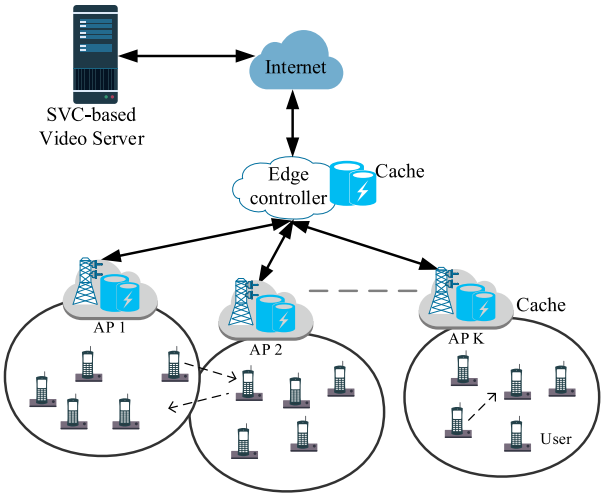

FIGURE 1. Scalable video caching system based on ICWN.

ICWN. In Section III, the hardness of the problem is analyzed, and a layered hierarchical caching scheme is designed by combining machine learning and optimization methods. A special case of the delay minimization problem is illustrated in Section IV, and an approximate solution is derived with 1/2 approximate ratio. In section V, through numerical simulations, the effectiveness of our proposed caching algorithms is verified. In Section VI, the work of this paper is concluded.

Notations of some important symbols are summarized in Table 1.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. NETWORK MODEL

As shown in Fig. 1, we consider a set $\mathcal{U}$ of mobile users, each of which is requesting SVC videos and moving in an area covered by ICWN. A set $\mathcal{K}$ of radio access points (APs) are located in the area and connected to a centralized edge controller (or edge router). The controller takes the role of aggregating SVC video traffic between the video server and APs. Both the edge controller and APs are equipped with caches. This architecture enables centralized optimization for content caching and cooperation in wireless access networks.

If a user in AP k requests a video which is neither cached at AP k nor at the edge controller, the video will be downloaded through Internet. Generally, the transmit delays from different network nodes are unequal. We denote by d_R, d_0 and d_k the average latency per bit incurred for a user associated with AP k to download videos from Internet, from the edge controller and from AP k, respectively.

We adopt the Gauss-Markov process to characterize users' mobility [22]. Let vectors $\ell_u(t) = (\ell_{ux}(t), \ell_{uy}(t))$ and $z_u(t) = \big(z_{ux}(t), z_{uy}(t)\big)$ respectively denote the location and velocity of user u at time t, where x and y represent the subscripts of two orthogonal components in a two-dimensional (2-D) area. The velocity in the next time slot is given by

$$z_u(t+1) = \varrho z_u(t) + (1-\varrho)\mu + \beta\sqrt{1-\varrho^2}w(t), \quad (1)$$

where ϱ is the memory parameter to reflect how current velocity affects future velocity, $\mu = (\mu_x, \mu_y)$ and β represent the central tendency and the dispersion of velocity, and $w(t)$ is an independent 2-D Gaussian process with zero mean and unit variance. The parameters μ, β and ϱ are assumed to be known a priori. The location of user u in the next time slot can be expressed as

$$\ell_u(t+1) = \begin{cases} \ell_u(t), & \text{if user is out of bounds,} \\ \ell_u(t) + z_u(t)\Delta T, & \text{else.} \end{cases} \quad (2)$$

When a user is outside the bounds, its location in the next time slot remains unchanged.

Based on the above Gauss-Markov process, the edge controller is aware of user mobility, and determines the association between users and APs in the next time slot. We assume that each user is associated with its closest AP. Let the binary decision variable a_{ku} indicate the association result, which is determined according to the following strategy:

$$a_{ku} = \begin{cases} 1, & k = \arg\max\limits_{k} \prod\limits_{i\in\mathcal{K}, i\neq k} \mathbb{P}(r_{ku}^2 < r_{iu}^2), \\ 0, & \text{else,} \end{cases} \quad (3)$$

where r_{ku} denotes the distance between user u and AP k, and the term $\prod_{i\in\mathcal{K}, i\neq k} \mathbb{P}(r_{ku}^2 < r_{iu}^2)$ is the probability that AP k is the nearest AP of user u.

Let $\mathcal{U}_k = \{u | u \in \mathcal{U}, a_{ku} = 1\}$ denote the set of users that will be connected to AP k. Assume each AP transmits video data using fixed power, and the bandwidth of AP k is equally allocated to the users in $\mathcal{U}_k$. The average downloading rate of user $u \in \mathcal{U}_k$ is calculated by

$$R_{ku} = \frac{W_k}{|\mathcal{U}_k|}\log_2\left(1 + \frac{P_k h_{ku}}{\sigma_0^2}\right), \quad (4)$$

where W_k and P_k denote the total available bandwidth and the transmit power of AP k. h_{ku} and σ_0^2 denote the channel fading and the noise power, respectively. Note that although inter-AP interference may exist, it can be well coordinated via different techniques [23]. Thus, for analytical simplicity, we assume that a user experiences a roughly static interference, which is similar as the model setting in [24], [25]. In addition, the average channel fading h_{ku} is given by $h_{ku} = \frac{A_r}{(r_{ku})^\tau}$, where A_r is a constant coefficient in large-scale fading model, and τ is the path loss exponent.

Next, we derive the cumulative distribution function (CDF) of R_{ku}. According to (1), in time slot $t+1$, the moving speed $z_u(t+1)$ of user u is a gaussian random variable with mean value $\xi_{zu} = \varrho z_u(t) + (1-\varrho)\mu$ and variance $\delta_z^2 = \beta^2(1-\varrho^2)$. Based on (2), the square of distance r_{ku}^2 is the sum squares of two independent gaussian variables, which follows a non-central chi-square distribution. Therefore, the probability density function (PDF) of r_{ku}^2 is given by

$$f_{r_{ku}^2}(r) = \frac{1}{2\delta^2} e^{-\frac{r+\xi_{ku}}{2\delta^2}} I_0\left(\frac{\sqrt{r\xi_{ku}}}{\delta^2}\right), \tag{5}$$

where $I_0(\cdot)$ is the zero-order modified Bessel function, $\delta^2 = \delta_z^2(\Delta T)^2$ denotes the variance of r_{ku}^2 and ξ_{ku} is the mean value given by

$$\xi_{ku} = \left(\ell_x^k - \ell_{ux}(t) - (\varrho z_{ux}(t) + (1-\varrho)\mu_x)\Delta T\right)^2 + \left(\ell_y^k - \ell_{uy}(t) - \left(\varrho z_{uy}(t) + (1-\varrho)\mu_y\right)\Delta T\right)^2,$$

where ℓ_x^k and ℓ_y^k are the horizontal and vertical coordinates of AP k. Finally, the cumulative distribution function (CDF) of R_{ku} can be derived based on (5), and is given by:

$$Q_{ku}(x) = \mathbb{P}(R_{ku} \le x) = \int_{A_k}^{+\infty} f_{r_{ku}^2}(r)dr, \tag{6}$$

where $A_k = \left(\frac{P_k A_r}{\sigma_0^2\left(2^{\frac{x|\mathcal{U}_k|}{W_k}} - 1\right)}\right)^{\frac{2}{\tau}}$ is derived from (4).

B. VIDEO DELIVERY MODEL

A set $\mathcal{V}$ of videos are stored in the remote server. Each video is encoded into L layers according to SVC protocol. Therefore, the video server can provide at most L versions for a requested video with different bitrates. Video v is divided into a set $\mathcal{M}_v$ of segments, each of which lasts for T_v seconds. Without loss of generality, we assume that T_v is equal to the duration of a time slot ΔT for analysis simplicity. The average segment size of the lth layer of video v is denoted by o_{vl}. Segment m of video v transmitted or cached in ICWN is refer to as an object and expressed by (v, m) in the following. If a user requests segment (v, m) with version l, the first l layers will be downloaded, and the transmit rate should be greater or equal to $\sum_{i=1}^{l} o_{vl}/T_v$. Moreover, each user has a buffer to store pre-fetched video data for subsequent use, and is capable of combining multiple layers into an integrated video. For example, a user requests a video with version l_2, but only layers from 1 to l_1 are cached locally, where $l_2 > l_1$. The user will obtain the first l_1 layers from local caches and download other layers through the backhaul link, and combine them together to form the requested video.

Moreover, let p_{kvl}^u indicate the probability that current channel condition of user u can afford the streaming of video v with version l, which is expressed by

$$p_{kvl}^u = \begin{cases} Q_{ku}\left(\sum_{i=1}^{2} \frac{o_{vi}}{T_v}\right), & l = 1 \\ Q_{ku}\left(\sum_{i=1}^{l+1} \frac{o_{vi}}{T_v}\right) - Q_{ku}\left(\sum_{i=1}^{l} \frac{o_{vi}}{T_v}\right), & l \in (l, L) \\ 1 - Q_{ku}\left(\sum_{i=1}^{L} \frac{o_{vi}}{T_v}\right), & l = L \end{cases} \tag{7}$$

Notice that if the transmit rate cannot afford the lowest video version (i.e. $R_{ku} < o_{v1}/T_v$), even the base layer of the SVC-encoded video cannot be successfully downloaded before playback, which will cause interruptions and rebuffering. However, the base layer will be still requested if the video session is not ended.

Similar as the video request model proposed in [26], we assume each request starts from the beginning of the video file and proceed sequentially. The viewing process is roughly divided into two phases: a browsing phase with high departure rate p_F and a viewing phase with low departure rate p_B. A viewing ratio of 15% is set as the boundary between the two phases. Meanwhile, since Zipf distribution has been established as a proper approximation to video popularity [27], we assume that if a user decides to switch to another video, the first segment of video v is requested with a probability given by:

$$q_v = \frac{v^{-\alpha}}{\sum_{i\in\mathcal{V}} i^{-\alpha}}, \tag{8}$$

where α is a parameter to determine the distribution skewness.

Based on the aforementioned model, if segment (v, m) is currently downloaded, segment (ν, n) will be requested in the next time slot with the following probability

$$\mathbb{P}\{(\nu, n)|(v, m)\} = \begin{cases} 1 - p_h, & \nu = v, n = m+1, m \le 0.15|\mathcal{M}_v|, \\ 1 - p_l, & \nu = v, n = m+1, m > 0.15|\mathcal{M}_v|, \\ \frac{p_F q_\nu}{1-q_v}, & \nu \ne v, n = 1, m \le 0.15|\mathcal{M}_v|, \\ \frac{p_B q_\nu}{1-q_v}, & \nu \ne v, n = 1, m > 0.15|\mathcal{M}_v|, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

C. PROBLEM FORMULATION

In this paper, we aim to minimize the average downloading delay of segments through effective caching schemes. We use

a binary decision variable b^m_{kvl} to indicate the caching decision of the l-th layer of segment (v, m), where $b^m_{kvl} = 1$ means layer l is cached in the edge controller ($k = 0$) or AP k ($k > 0$) and $b^m_{kvl} = 0$, otherwise. Any requested contents that are not cached at the mobile edge should be downloaded through Internet. For user $u \in \mathcal{U}_k$ requesting segment (v, m) with version l, the average delay to obtain all layers from 1 to l is given by:

$$\begin{aligned} d^m_{kvl} &= \sum_{i=1}^{l} d_k o_{vi} b^m_{kvi} + d_0 o_{vi}(1 - b^m_{kvi}) b^m_{0vi} \\ &\quad + d_R o_{vi}(1 - b^m_{kvi})(1 - b^m_{0vi}) \\ &= \sum_{i=1}^{l} o_{vi}[d_R + (d_R - d_0) b^m_{kvi} b^m_{0vi} - (d_R - d_k) b^m_{kvi} \\ &\quad - (d_R - d_0) b^m_{0vi}]. \end{aligned} \tag{10}$$

Moreover, let λ^{um}_{kv} denote the probability of user $u \in \mathcal{U}_k$ requesting video segment (v, m) in the next time slot, which can be predicted according to (9). Based on (7) and (10), the total average delay of all mobile users can be expressed as:

$$D_{total} = \sum_{k\in\mathcal{K}} \sum_{u\in\mathcal{U}_k} \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} \lambda^{um}_{kv} p^u_{kvl} d^m_{kvl}. \tag{11}$$

To minimize D_{total}, we formulate the following optimization problem:

$$\begin{aligned} \min_{\{b^m_{kvl}\}} & \sum_{k\in\mathcal{K}} \sum_{u\in\mathcal{U}_k} \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} \lambda^{um}_{kv} p^u_{kvl} d^m_{kvl} \\ \text{s.t.} & \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} b^m_{kvl} o_{vl} \le C_k, \forall k \in \mathcal{K} \cup \{0\}, \\ & b^m_{kvl} \in \{0, 1\}, \forall k \in \mathcal{K} \cup \{0\}, \end{aligned} \tag{12}$$

where C_k and C_0 denote the cache capacities of AP k and the edge controller, respectively. The first constraint in (12) indicates that the total amount of contents in each cache device should not exceed the corresponding cache capacity.

III. LAYERED HIERARCHICAL CACHING FOR SVC-BASED VIDEO STREAMING

In this section, we first prove the NP-hardness of problem (12), then simplify the problem and propose a layered hierarchical caching algorithm for SVC-based video streaming over ICWN.

Theorem 1: Problem (12) *is NP-hard.*

Proof: Knapsack problem (KP) is a well known NP-Hard problem. We prove Theorem 1 by reducing KP to Problem (12). In KP, we are given a set of items and a knapsack. These items have different weights and values, and can be packed into the knapsack which has limited weight capacity. The objective is to pick out part of the items to make sure that their total value is the largest while their total weight is less or equal to the knapsack capacity. Note that KP is a special case of problem (12) if $C_k = 0, \forall k \in \mathcal{K}$, i.e., all APs are not cache-enabled. In this case, each layer of a segment can be considered as an item in KP, where the weight and the value of each item are o_{vl} and $o_{vl}(d_R - d_0) \sum_{k'\in\mathcal{K}} \sum_{u\in\mathcal{U}_k} \sum_{i=l}^{L} p^u_{k'vi} \lambda^{um}_{k'v}$, respectively. The above steps of reduction can be finished within polynomial time, which proves the NP-hardness of problem (12). □

A. PROBLEM REFORMULATION

According to Theorem 1, it is hard to obtain the optimal solution of problem (12) within polynomial time. Therefore, we first simplify the problem according to the theory of KP, and then propose a heuristic algorithm.

We rewrite (11) as

$$D_{total} = D - \sum_{k\in\mathcal{K}\cup\{0\}} \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} D^m_{kvl} b^m_{kvl}, \tag{13}$$

where D is a constant value given by:

$$D = \sum_{k\in\mathcal{K}} \sum_{u\in\mathcal{U}_k} \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} \lambda^{um}_{kv} p^u_{kvl} \sum_{i=1}^{l} o_{vi} d_R, \tag{14}$$

and

$$D^m_{kvl} = \begin{cases} o_{vl}\,(d_R - d_0) \sum_{k'\in\mathcal{K}} \sum_{u\in\mathcal{U}_k} \sum_{i=l}^{L} p^u_{k'vi} \lambda^{um}_{k'v} \left(1 - b^m_{k'vl}\right), \\ \qquad\qquad\qquad\qquad k = 0, \\ o_{vl} \sum_{u\in\mathcal{U}_k} \sum_{i=l}^{L} p^u_{kvi} \lambda^{um}_{kv} \left[(d_R - d_k) - (d_R - d_0)\, b^m_{0vl}\right], \\ \qquad\qquad\qquad\qquad k \neq 0. \end{cases} \tag{15}$$

To minimize the delay D_{total} is equivalent to solving the following optimization problem:

$$\begin{aligned} \max_{b^m_{kvl}} & \sum_{k\in\mathcal{K}\cup\{0\}} \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} D^m_{kvl} b^m_{kvl} \\ \text{s.t.} & \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} b^m_{kvl} o_{vl} \le C_k, \quad \forall k \in \mathcal{K} \cup \{0\}, \\ & b^m_{kvl} \in \{0, 1\}, \quad \forall k \in \mathcal{K} \cup \{0\}. \end{aligned} \tag{16}$$

Problem (16) can be interpreted as a knapsack problem. The cache devices at the edge controller and APs play the role of knapsacks in KP. Accordingly, there are a total of $|\mathcal{K}| + 1$ knapsacks with capacities $C_0, C_1, \cdots, C_K$, respectively. Each layer of a video segment corresponds to an item in KP. Therefore, all $\sum_v |\mathcal{V}| \cdot L \cdot |\mathcal{M}_v|$ layers form an item set $\mathcal{O}$. Each item can be put into more knapsacks. For the l-th layer of video segment (v, m), its weight is o_{vl}, but its value D^m_{kvl} is only determined after it has been put into a knapsack. The objective is to pick out $K + 1$ sets of items which maximize the total value.

Algorithm 1 Layered Hierarchical Caching Scheme

Input: $C_k, \lambda_{kv}^{um}, o_{vl}, p_{kvl}^{u}$
Output: $\mathcal{S}_k, \forall k \in \mathcal{K} \cup \{0\}$

Parameter Initialize.
(1) Map each layer of video segments into an item set $\mathcal{O}$ with weight o_{vl}.
(2) Set $\mathcal{S}_k = \Phi$, where $k \in \mathcal{K} \cup \{0\}$.
Determine the caching priority for each item based on SVM.
repeat
  Update the values of items in $\mathcal{O}$ for knapsack 0.
  Call Algorithm 2 for knapsack 0 to obtain $\mathcal{S}_0$.
  for $k \in \mathcal{K}$ **do**
    Update the values of items in $\mathcal{O}$ for knapsack k.
    Call Algorithm 2 for knapsack k to obtain $\mathcal{S}_k$.
  end for
until
Maximal $\sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}_k)$ is obtained, or the number of iterations exceeds the maximal threshold I.
return $\mathcal{S}_k$

B. LAYERED HIERARCHICAL CACHING ALGORITHM

Layered video contents can be cached in both the edge controller and an AP simultaneously. According to (15), the value of an item D_{kvl}^{m} may vary when the item is put into different knapsacks. Specifically, the value of an item in the edge controller depends on whether it has been cached in APs. Similarly, the value of an item in an AP depends on whether it has been placed in the edge controller.

Therefore, the caching priority has a significant impact on the performance, which should be determined by considering both network parameters and popularity of video segments. However, it is hard to obtain an exact priority directly for each item. Existing solutions such as branch-and-bound algorithms may suffer from forbidding computational complexity or unsatisfactory optimality [28]. To overcome the above disadvantages, some machine learning (ML) based approaches have been widely adopted and proved to be effective [29]. Therefore, in this paper, we determine the priority based on ML, which will be detailed in Section III-C.

Based on the caching priority, we propose a heuristic layered hierarchical caching scheme, which is given as Algorithm 1. The details of each step are described as follows:

Step 1: Initialize the selected item set $\mathcal{S}_k$ for each knapsack k, where $k \in \mathcal{K} \cup \{0\}$. Let $g_v(\cdot)$ and $g_w(\cdot)$ denote the value function and the weight function of an item or a set, respectively. We can obtain that $g_v(\Phi) = 0$ and $g_w(\Phi) = 0$, where Φ denotes an empty set.

Step 2: During the online decision making phase, the instant parameters of items and networks within our considered region are readily available at the controller. We can obtain the caching priority for each item in real time based on the well trained SVM model.

Step 3: For knapsack 0, the value of each item D_{0vl}^{m} in set $\mathcal{O}$ is updated. These items are sorted by the decreasing order of unit value D_{0vl}^{m}/o_{vl}, and as many items as possible are selected according to the order until knapsack 0 cannot be filled with more items. At this time, a set $\mathcal{S}_0$ of videos cached by the edge controller is obtained, and the corresponding b_{0vl}^{m} is updated. After that, the values of items is updated for each AP. Similarly, a set $\mathcal{S}_k$ of selected items are cached in AP k, and the b_{kvl}^{m} is determined accordingly.

Step 4: Based on the obtained $\mathcal{S}_0$ and $\mathcal{S}_k$ in Step 3, the sum value $\sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}_k)$ can be calculated. If the increment of the above value is less than a constant tolerance or the number of iterations exceeds the maximal threshold I, the item sets for all knapsacks are obtained. Otherwise, the next iteration of the algorithm starts from Step 2.

The single knapsack filling algorithm applied in Step 3 is shown in Algorithm 2. Notice that our proposed layered hierarchical cache scheme can be executed with low computational overhead and finished within polynomial time, which can be implemented in the centralized edge controller.

Algorithm 2 Single Knapsack Filling Algorithm

Input: $\mathcal{O}, C_k, \lambda_{kv}^{um}, o_{vl}, p_{kvl}^{u}$
Output: $\mathcal{S}_k$

Calculate D_{kvl}^{m} according to (15).
Sort the items of $\mathcal{O}$ in decreasing unit-value order according to D_{kvl}^{m}/o_{vl}.
for each item a_i in $\mathcal{O}$, $1 \le i \le |\mathcal{O}| - 1$ **do**
  if $g_w(\mathcal{S}_k \cup \{a_i\}) \le C_0$ **then**
    Set $\mathcal{S}_k := \mathcal{S}_k \cup \{a_i\}$.
  end if
end for
return $\mathcal{S}_k$

C. CACHING PRIORITY DETERMINATION BASED ON MACHINE LEARNING

First, a sufficiently large number of training data have to be generated. Therefore, we randomly simulate multiple scenarios with various user requests and network parameters. For each scenario, a specific problem (12) is obtained accordingly. Ideally, if these problems can be solved optimally, an accurate labeled training data set will be formulated based on the solutions. Specifically, if $b_{0vl}^{m} = 1$, the controller will get the priority to cache the lth layer of segment (v, m), and the corresponding item in $\mathcal{O}$ is labeled as 1; otherwise, the item is labeled as 0. After collecting all training labels from multiple scenarios, the training set becomes large enough to make a machine learning model be well trained.

However, problem (12) is hard to be solved optimally within a reasonable time, since the complexity of searching b_{0vl}^{m} exhaustively is as high as $2^{|\mathcal{O}|}$, which is unaffordable in practice when $|\mathcal{O}|$ is very large. Therefore, in the following, we design a novel exhausting search method with sublinear reduction [30] to speed up the generation of training data.

The essential idea is to effectively reduce the size of $\mathcal{O}$ in each scenario. We first remove tail items with extremely low or zero unit-values, and sample a subset of the residual

items to obtain a reduced subset $\mathcal{O}'$. Then, we search the optimal b^m_{0vl} in set $\mathcal{O}'$ exhaustively. An item that is not included in $\mathcal{O}'$ shares the priority of its most similar item in $\mathcal{O}'$. Specifically, we sort the items in $\mathcal{O}$ in decreasing unit-value order according to D^m_{kvl}/o_{vl}, and obtain the ranks κ_{ik} for each item $i \in \mathcal{O}$ in AP $k \in \mathcal{K}$ and the controller. These ranks are further normalized for similarity comparison, and the caching capacities $C_k (k \in \mathcal{K} \cup \{0\})$ are reduced by the same sampling ratio. Each record of the training set includes the normalized unit-value ranks, weight, d_R, d_0, d_k, C_0, C_k, α, and a corresponding label of caching priority. Although the exhausting algorithm cannot be executed in real time, it does not impair the effectiveness of our proposed method, because the training data are generated offline.

In addition, we design a binary classifier based on support vector machine (SVM), which takes the training data as input and outputs the caching priority for each item. Intuitively, the unit-value ranks of an item in different caches are closely related to its request characteristics, and are suitable to be utilized as training features. Moreover, we observe that their mean value reflects the average popularity of an item, and their variance indicates whether an item is uniformly requested by all users in different APs. Therefore, we adopt all the above elements as training features, i.e. d_R, d_0, d_k, C_0, C_k, α and κ_{ik}, together with the corresponding means and variances of κ_{ik}. Note that, the caching capacities of the controller and each AP are reduced by the same sampling proportion accordingly.

To reduce the computational complexity, we choose radial basis function (RBF) as the kernel function of SVM [31], which can map the samples nonlinearity to a higher dimensionality space. Note that such offline training process does not take up the time of online decision making and the well-trained model can be used directly during the service time.

D. COMPLEXITY OF ALGORITHM 1

The complexity of our proposed method is analyzed. Since the training process can be performed offline, we only consider the online decision making phase of Algorithm 1. First, we sort the items in $\mathcal{O}$ in decreasing unit-value order in each knapsack, whose complexity is $O((K+1)|\mathcal{O}|\log|\mathcal{O}|)$. Then, for each item, the complexity of judging an item's priority based on the SVM model is $O(N_c d_{in})$ [32], where N_c is the number of output categories, i.e. 2 in our model, and d_{in} represents the dimension of the input vectors. The complexity of Algorithm 2 is $O(|\mathcal{O}|\log|\mathcal{O}|)$. The maximal computational complexity of Algorithm 1 can be calculated by $O(N_c d_{in}|\mathcal{O}| + I(K+1)|\mathcal{O}|\log|\mathcal{O}|)$.

IV. A SPECIAL CASE: REMOTE DOWNLOAD DELAY IS MANY-FOLD HIGHER THAN LOCAL DOWNLOAD DELAY

In practice, the delay to download a video from remote server is generally many-fold higher than that from local caches at the network edge [33]. In this section, we consider a special case of Problem (12), where the condition $d_R - d_k \approx d_R - d_0 \approx d_R$ is satisfied.

Based on the above mentioned approximation, the average delay given in (10) can be simplified as

$$\widetilde{d}^m_{kvl} = d_R \sum_{i=1}^{l} o_{vi}(1 - b^m_{kvi})(1 - b^m_{0vi}). \tag{17}$$

Moreover, let

$$q^m_{kvl} = \frac{\widetilde{d}^m_{kvl}}{d_R} = \sum_{i=1}^{l} o_{vi}\left(1 - b^m_{kvi}\right)\left(1 - b^m_{0vi}\right), \tag{18}$$

where q^m_{kvl} denotes the amount of layers from 1 to l of video v, which are neither cached in AP k nor in the edge controller. Correspondingly, the delay minimization problem (12) can be equivalently expressed as

$$\begin{aligned} \min_{b^m_{kvl}} \; & R_{miss} = \sum_{k\in\mathcal{K}} \sum_{u\in\mathcal{U}_k} \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} \lambda^{um}_{kv} p^u_{kvl} q^m_{kvl} \\ \text{s.t.} \; & \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} b^m_{kvl} o_{vl} \le C_k, \quad \forall k \in \mathcal{K}\cup\{0\}, \\ & b^m_{kvl} \in \{0,1\}, \quad \forall k \in \mathcal{K}\cup\{0\}. \end{aligned} \tag{19}$$

where R_{miss} means the video data downloaded from remote server due to cache miss.

We define the cache hit ratio of our considered ICWN as the percentage of requests that can be retrieved from the edge controller or an AP. Obviously, solving Problem (19) is equivalent to maximizing the cache hit ratio. Therefore, Problem (19) is also referred to as cache hit ratio maximization problem in this paper.

Although we can adopt Algorithm 1 to obtain a heuristic solution since Problem (19) is a special case of Problem (12), we would like to design a more efficient algorithm with low complexity and guaranteed approximation ratio by utilizing the special structure of Problem (19).

A. SIMPLIFIED LAYERED HIERARCHICAL CACHING ALGORITHM

Lemma 1: In the optimal solution of problem (19)*, no layers of the SVC videos can be both cached simultaneously by the edge controller and an AP, i.e.* $b^m_{kvl} b^m_{0vl} = 0$ *or* $b^m_{kvl} + b^m_{0vl} \le 1, \forall k \in \mathcal{K}, \forall v, \forall l, \forall m$.

Proof: We prove Lemma 1 by contradiction. Suppose there is an optimal solution that caches the l-th layer of video segment (v, m) both in the edge controller and AP k, i.e. $b^m_{kvl} = 1$ and $b^m_{0vl} = 1$. Obviously, the objective value of (19) will not be changed if we set $b^m_{kvl} = 0$. In other words, removing the cached layer of (v, m) in AP k would have no impact on the result. Then, we can fill the caching space of AP k that previously cached the removed content with other uncached contents, which can certainly improve the optimal solution. This contradicts the assumption of the optimality. □

Based on Lemma 1, we further simplify R_{miss} by

$$R_{miss} = B - \sum_{k\in\mathcal{K}\cup\{0\}} \sum_{v\in\mathcal{V}} \sum_{l=1}^{L} \sum_{m\in\mathcal{M}_v} R^m_{kvl} b^m_{kvl}, \tag{20}$$

where B is a constant value given by

$$B = \sum_{k\in\mathcal{K}}\sum_{u\in\mathcal{U}_k}\sum_{v\in\mathcal{V}}\sum_{l=1}^{L}\sum_{m\in\mathcal{M}_v} p^u_{kvl}\lambda^{um}_{kv}\sum_{i=1}^{l} o_{vi}, \tag{21}$$

and

$$R^m_{kvl} = \begin{cases} o_{vl}\sum\limits_{k'\in\mathcal{K}}\sum\limits_{u\in\mathcal{U}_k}\sum\limits_{i=l}^{L} p^u_{k'vi}\lambda^{um}_{k'v}, & k=0, \\ o_{vl}\sum\limits_{u\in\mathcal{U}_k}\sum\limits_{i=l}^{L} p^u_{kvi}\lambda^{um}_{kv}, & k\in\mathcal{K}. \end{cases} \tag{22}$$

Therefore, Problem (19) can be equivalently transformed into the following problem

$$\begin{aligned} \max_{b^m_{kvl}} \quad & \sum_{k\in\mathcal{K}\cup\{0\}}\sum_{v\in\mathcal{V}}\sum_{l=1}^{L}\sum_{m\in\mathcal{M}_v} R^m_{kvl} b^m_{kvl} \\ \text{s.t.} \quad & \sum_{v\in\mathcal{V}}\sum_{l=1}^{L}\sum_{m\in\mathcal{M}_v} b^m_{kvl} o_{vl} \le C_k, \quad \forall k\in\mathcal{K}\cup\{0\}, \\ & b^m_{kvl} + b^m_{0vl} \le 1, \quad \forall k\in\mathcal{K},\ \forall v,\ \forall l,\ \forall m, \\ & b^m_{kvl}\in\{0,1\}, \quad \forall k\in\mathcal{K}\cup\{0\},\ \forall v,\ \forall l, \forall m. \end{aligned} \tag{23}$$

Similarly as what we did for solving Problem (16) in Section III, Problem (23) can also be interpreted as a knapsack problem. Here, for the l-th layer of video segment (v, m), the weight and the value are respectively o_{vl} and R^m_{kvl} when it is put into knapsack k. A new constraint is added that an item cannot exist both in knapsack 0 and knapsack k, where $k \in \mathcal{K}$.

Since the edge controller and APs have similar downloading delay compared with remote video server, the priority of the edge controller is higher than APs, then all videos should be sorted according to D^m_{0vl} (which is not essentially different from R^m_{0vl} here), and packed with knapsack 0 to get a set of videos as $\mathcal{S}_0$. Then, the value D^m_{kvl} of all videos in the AP (which is not essentially different from R^m_{kvl}) is updated. For the video files in $\mathcal{S}_0$, their value in each AP is updated to 0. That is to say, knapsack 0 always has highest priority to cache video contents in the special case.

Based on the above analysis, we propose an approximal method as shown in Algorithm 3. First, we sort the items of $\mathcal{O}$ in decreasing unit-value order, i.e. $\frac{g_v(a_i)}{g_w(a_i)} \ge \frac{g_v(a_j)}{g_w(a_j)}$ if $i \le j$, where a_i and a_j denote the ith and the jth items in $\mathcal{O}$. Then, we put as many items as possible according to this order into knapsack 0, and obtain a item set of $\mathcal{S}_0$. Second, for each knapsack k ($k \in \mathcal{K}$), sort the items in $\mathcal{O}\backslash\mathcal{S}_0$ in the same decreasing unit-value order. Fill knapsack k with as many items as possible according to the aforementioned decreasing order.

B. APPROXIMATION RATIO

To show the performance of our proposed scheme, we estimate its approximation ratio compared with the optimal solution.

Algorithm 3 Heuristic Caching Scheme for the Special Case

Input:
$C_k, \lambda^{um}_{kv}, o_{vl}, p^u_{kvl}$,
Output:

```
Map all layers of videos into an item set O with weight o_vl. Calculate R^m_kvl according to (22). Set S_k = Φ, ∀k ∈ K ∪ {0}, where Φ denotes an empty set.
Sort the items of O in decreasing unit-value order according to R^m_0vl.
for each item a_i in O, 1 ≤ i ≤ |O| − 1 do
    if g_w(S_0 ∪ {a_i}) ≤ C_0 then
        Set S_0 := S_0 ∪ {a_i}.
    end if
end for
for k ∈ K do
    Set O_k = O\S_0
    Sort items of O_k in decreasing unit-value order according to R^m_kvl.
    for each item a_j in O_k, 1 ≤ j ≤ |O_k| − 1 do
        if g_w(S_k ∪ {a_j}) ≤ C_k then
            Set S_k := S_k ∪ {a_j}.
        end if
    end for
end for
return S_k, ∀k ∈ K ∪ {0}
```

Theorem 2: Algorithm 3 can provide at least 1/2 *approximations to the optimal solution.*

Proof: Assume there is an optimal algorithm for solving Problem 23. By adopting the optimal algorithm and Algorithm 3, we obtain two results as $\mathcal{S}^{opt}_k$ and $\mathcal{S}_k$, which indicate the items finally selected for knapsack k. Obviously, if local caches' capacities in considered ICWN are large enough to cache all versions of videos in library, our proposed scheme can achieve the optimal performance. However, in general, the local caches cannot cache all videos. In this case, the proof is given as follows.

It is assumed that the items in $\mathcal{O}$ have been sorted in decreasing unit-value order. Let a_i be the first excluded item when filling $\mathcal{S}_0$, i.e. $g_w(\{a_1, \ldots, a_{i-1}\}) \le C_0$ and $g_w(\{a_1, \ldots, a_i\}) > C_0$. Let $\mathcal{S}^\dagger_0 := \{a_1, \ldots, a_{i-1}\}$ and $\mathcal{S}^\ddagger_0 := \mathcal{S}^\dagger_0 \cup \{a_i\}$. Similarly, for each knapsack k ($k \in \mathcal{K}$), sort the items in $\mathcal{O}\backslash\mathcal{S}^\ddagger_0$ in decreasing unit-value order according to their values in knapsack k. Let c^j_k be the first excluded item for knapsack k. Set $\mathcal{S}^\dagger_k := \{c^1_k, \ldots, c^{i-1}_k\}$ and $\mathcal{S}^\ddagger_k := \mathcal{S}^\dagger_k \cup \{c^j_k\}$. Obviously, $g_w(\mathcal{S}^\ddagger_k) > C_k, \forall k \in \mathcal{K}\cup\{0\}$.

If the binary constraint that $b^m_{kvl} \in \{0, 1\}$ is relaxed to $b^m_{kvl} \in [0, 1]$, (23) can be considered as a linear programming problem, whose optimal objective value is an upper bound of $\sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}^{opt}_k)$. If we expand the capacity of knapsack k to $g_w(\mathcal{S}^\ddagger_k)$ by solving the linear programming problem, we can easily obtain that $\mathcal{S}^\ddagger_k$ is the optimal solution. Thus, we have $\sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}^\ddagger_k) \ge \sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}^{opt}_k)$.

Let $a_0^{max} \in C_0$ be a item with maximum value for knapsack 0, where $g_w(a_0^{max}) \leq C_0$ is satisfied. Let $a_k^{max} \in \mathcal{O}\backslash\{a_0^{max}\}$ be an item with maximum value for knapsack k, which is constrained by $g_w(a_k^{max}) \leq C_k, k \in \mathcal{K}$.

With the above analysis, we can derive that

$$\begin{aligned}\sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}_k) &= \max\left(\sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}_k), \sum_{k\in\mathcal{K}\cup\{0\}} g_v(a_k^{max})\right)\\ &\geq \max\left(\sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}_k^{\dagger}), \sum_{k\in\mathcal{K}\cup\{0\}} g_v(a_k^{max})\right)\\ &\geq \frac{1}{2}\left(\sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}_k^{\dagger}) + \sum_{k\in\mathcal{K}\cup\{0\}} g_v(a_k^{max})\right)\\ &= \frac{1}{2}\sum_{k\in\mathcal{K}\cup\{0\}} g_v\left(\mathcal{S}_k^{\dagger}\cup\{a_k^{max}\}\right)\\ &\geq \frac{1}{2}\sum_{k\in\mathcal{K}\cup\{0\}} g_v\left(\mathcal{S}_k^{\ddagger}\right) \geq \frac{1}{2}\sum_{k\in\mathcal{K}\cup\{0\}} g_v(\mathcal{S}_k^{opt}).\end{aligned}$$

In summary, our proposed algorithm can achieve at least 1/2 of the optimal total value. □

C. COMPLEXITY OF ALGORITHM 3

In Algorithm 3, all items in $\mathcal{O}$ are first sorted for the centralized controller, and then K similar sorting processes are performed over $\mathcal{O}_k$ for all APs. Therefore, the complexity of entire algorithm can be given by $O(|\mathcal{O}|\log|\mathcal{O}| + \sum_{k\in\mathcal{K}}|\mathcal{O}_k|\log|\mathcal{O}_k|)$. Compared with Algorithm 1, Algorithm 3 is of low complexity and easy to be implemented.

V. NUMERICAL RESULTS

In this section, the performance of our proposed caching schemes are evaluated. Our simulation platform is built based on MATLAB. In the platform, user movement and video request are simulated and predicted according to the models in Section II, and the SVM classifier tool integrated in MATLAB is adopted. Average transmission delay of downloading a video segment, cache hit rate and QoE of users are adopted as main metrics.

Network Setting: We consider an ICWN with 10 APs which are connected to and controlled by an edge controller. Both the controller and each AP are equipped with caching devices with size of 5 GB and 1 GB, respectively. We set the available bandwidth, the transmit power and the noise power to be $W_k = 20$ MHz, $P_k = 35$ dBm and $\sigma_0^2 = -105$ dBm, respectively. The number of mobile users in the ICWN is 100 in default. The user mobility is characterized by Gauss-Markov process, where the average velocity is $\mu = (0.6, 3.8)$ m/s, and other mobility parameters are set to be $\beta = 2$ and $\varrho = 0.8$, respectively. The latency parameters are $d_R = 80$ ns, $d_0 = 30$ ns, and $d_k = 20$ ns for each AP, respectively.

Application Setting: Each video in the server is encoded into 1 base layer and 4 enhancement layers. Normally, the bitrate of video layer depends on the encoding parameters and the content of the video. For the purpose of simplifying simulation, it is assumed that different videos have similar bitrates, and each layer has a bitrate of 500 Kbps. Therefore, with 500 Kbps in step, the video server offers 5 versions of video, ranging from 500 Kbps to 2500 Kbps, respectively. The departure rates p_F and p_B are set to 0.7 and 0.3, respectively. The duration of a video follows a uniform distribution from 2 minutes to 10 minutes. A video is divided into multiple segments, each of which lasts for 2 seconds.

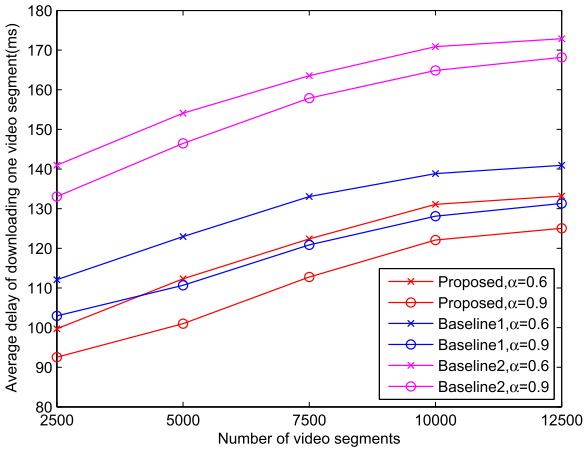

FIGURE 2. Average delay of downloading one video segment of different caching schemes.

Comparison Baseline: We compare the performance of our proposed schemes with the following two baselines.

- **Baseline 1**: The first one is a non-layered caching scheme. Different from our proposed methods, *non-layered caching* approaches use traditional non-scalable coding protocols to encode videos. Typically, SVC consumes approximately 20% more bits to achieve the same video quality, which is an additional overhead of adopting layered coding schemes [34].
- **Baseline 2**: The caching method proposed in [10] is also simulated as a comparison baseline, which caches each video as a whole file including all video segments.

A. AVERAGE TRANSMISSION DELAY

Fig. 2 depicts the delay performance of different caching schemes with the increasing number of video segments. The parameter α of Zipf distribution is set to 0.6 and 0.9, respectively. As shown in Fig. 2, for all caching schemes, the larger the number of video segments, the higher the average delay of downloading a single video segment. The reason that our proposed schemes outperform the baselines is that the *layered cache* schemes reuse low-layer video data, which fully explore the benefits of layered video encoding. Fig. 2 shows that the content reusing gain can overcome the 20% overhead of layered coding and leads to better performance. Moreover, by caching each video as a whole file, the performance of baseline 2 is inferior to that of the others which cache video contents at segment level. This is because baseline 2 is not

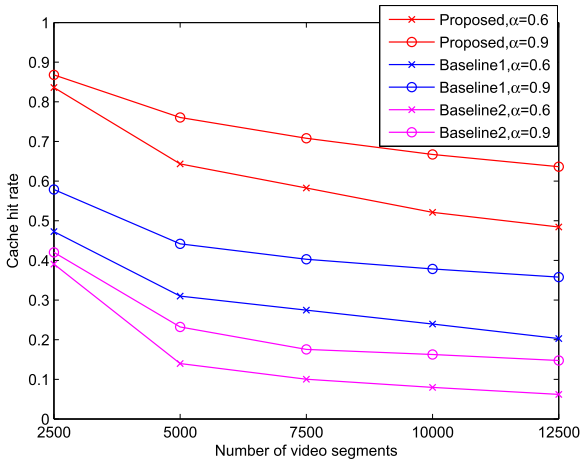

FIGURE 3. Cache hit rates of different caching schemes.

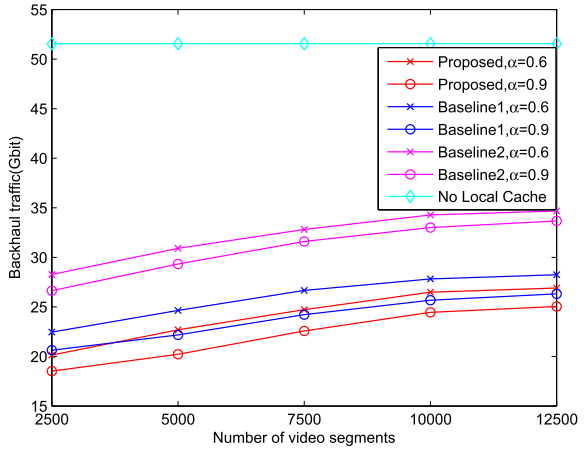

FIGURE 4. Backhaul traffic load of different caching schemes.

mobility aware. When a user is moving from one AP to another, a large number of pre-cached video segments in the user's previous associated AP may become useless. In addition, when α increases from 0.6 to 0.9, i.e. video requests become more concentrated, the average delays of all schemes are reduced, which is consistent with the characteristics of Zipf distribution. The more concentrated the video requests are, the better performance will be obtained.

B. CACHE HIT RATE

Fig. 3 shows the trend of cache hit rate under different cache schemes, α values and numbers of video segments. As the number of video segments increases, the cache hit rates of all caching schemes decrease. However, the performance of our proposed *layered cache* scheme is better than that of other baselines. It is noted that, benefiting from the data reuse feature of layered video, the cache hit rate of the *layered cache* scheme with $\alpha = 0.6$ is higher than that of the *non-layered caching* scheme with $\alpha = 0.9$. What's more, our proposed cache scheme and baseline 1 are both better than baseline 3 thanks to the former two being able to use cache space more efficiently.

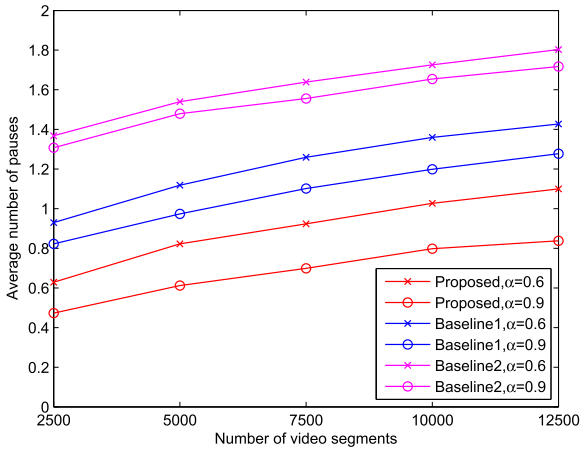

FIGURE 5. Average number of pauses under different caching schemes.

C. BACKHAUL TRAFFIC LOAD

Fig. 4 shows the backhaul traffic load per 5 mins under different caching schemes. The scheme named *no local cache* means that both the edge controller and APs are not equipped with cache devices. Comparing with the *no local cache* scheme, the proposed *layered cache* scheme can effectively reduce the backhaul traffic load by up to 65%. It shows that caching is an effective solution to alleviate the bandwidth pressure on the backhaul link. The *layered cache* scheme performs better than baselines in different video quantities. The properties of different α also conform to the Zipf distribution.

D. AVERAGE NUMBER OF PAUSES

Since viewing interruption largely affects the QoE of users, we simulate the average number of pauses for different schemes. The numerical results are averaged over a large number of independent runs each with a duration of 5 minutes. As shown in Fig. 5, the average numbers of pauses under different caching schemes increase with the number of video segments. Compared with baselines, users experience less number of pauses when adopting our proposed caching scheme. In addition, when user requests are more concentrated, i.e. the parameter α of Zipf distribution is changed from 0.6 to 0.9, the number of pauses becomes less accordingly.

E. IMPACT OF NUMBER OF USERS

Fig. 6 depicts the impact of user scales on our proposed caching schemes. The total number of users is set to 60, 80, 100, and 120, respectively. The parameter α in Zipf distribution is set to 0.6. In our design, most performance metrics depend on the total number of users. According to (6) and (7), with the increasing number of users, the download rate tends to decrease, resulting in more requests of videos with lower version. However, although the average video quality is impaired, the performance of the average delay, the cache hit rate, the backhaul load and the number of pauses is improved. This is because more users are probable to request for videos with low qualities, thus the distribution of requests becomes

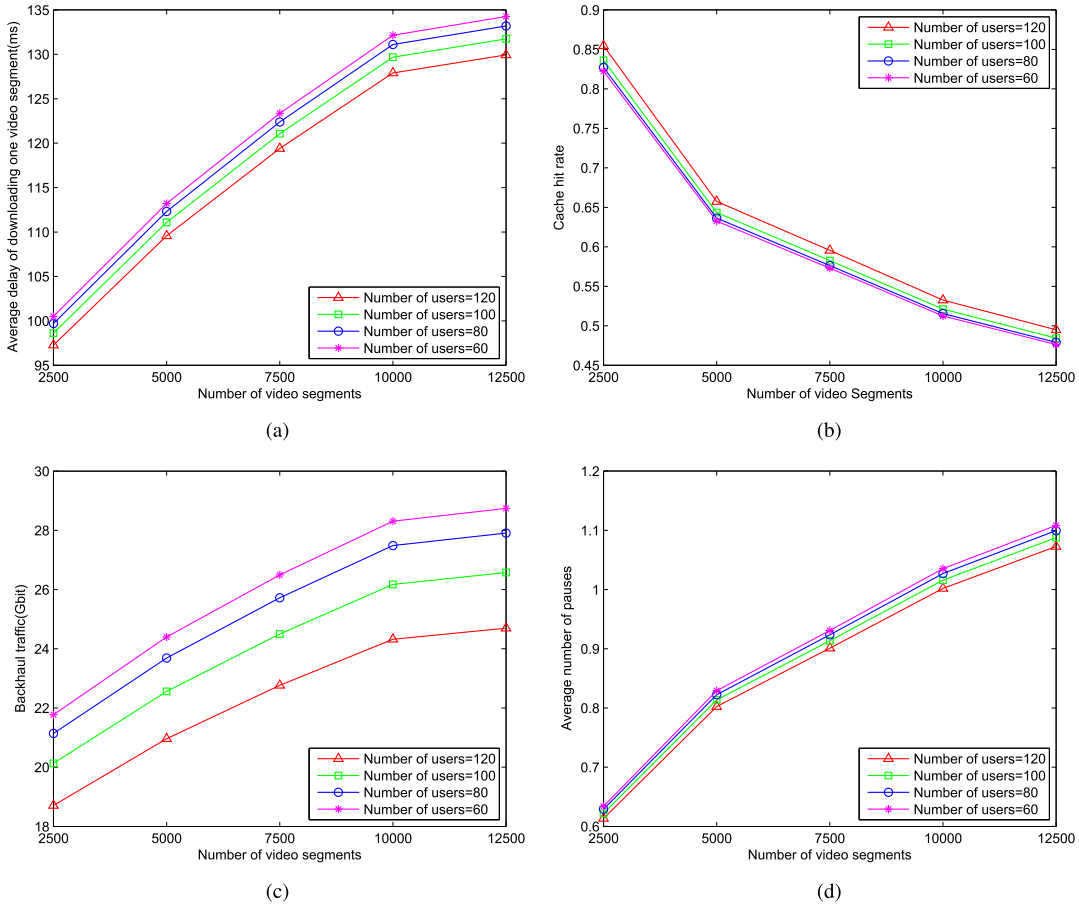

FIGURE 6. Performance comparison of our proposed caching scheme with respect to different user scales: (a) average delay of downloading one video segment; (b) cache hit rate; (c) backhaul traffic load; (d) average number of pauses.

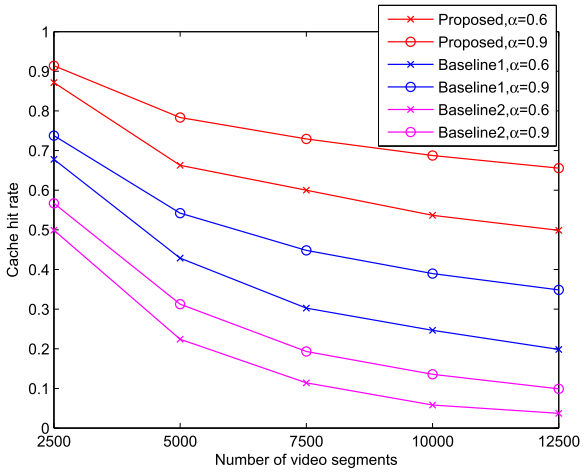

FIGURE 7. Cache hit rates of different caching schemes in special case.

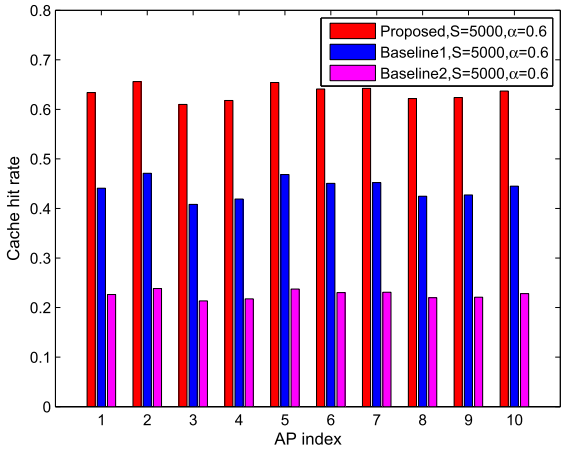

FIGURE 8. Cache hit rates in different APs.

concentrated. More requested videos can be found in local caches. Therefore, higher cache hit rate and lower backhaul load are observed. Furthermore, since lower versions of video includes less bits and the average delay per bit is constant in the simulation, there will be lower average delay and less potential pauses if less bits are downloaded.

F. SIMULATION RESULTS OF THE SPECIAL CASE

In the special case introduced in Section IV, where $d_R - d_k \approx d_R - d_0 \approx d_R$ is satisfied, minimizing the system total latency is equivalent to maximizing the cache hit rate. As shown in Fig. 7, the trend of the curve distribution of this special case is similar to the general delay minimization case. Other

analysis of the proposed layered cache scheme and the other two baseline schemes have been respectively presented in Fig. 2, Fig. 3, Fig. 4, Fig. 5, and Fig. 6 will not be discussed here.

Fig. 8 shows the cache hit rate for three schemes in different APs, where the number of video segments is set to 5000, and the parameter α of Zipf distribution is set to 0.6. The cache hit rate in different APs varies due to diverse interests of users in different APs for video contents. By adopting our proposed caching scheme, users are more probable to obtain higher cache hit rate in each AP.

VI. CONCLUSION

In this paper, we address the caching problem for SVC-based video steaming over ICWN, which is characterized by both layered video contents and hierarchical network structures. We formulate a 0-1 programming problem to minimize the average video transmission latency. The NP-hardness of the problem is proved, and we simplify it into a special KP for the ease of problem solving. A layered hierarchical caching scheme is proposed to solve the problem within polynomial time. In addition, we consider a special but common case where the remote download delay is many-fold higher than the local download delay. The original problem can be equivalently simplified into a cache hit rate maximization problem, and an algorithm is proposed to solve the simplified problem with a $1/2$ approximation ratio. Simulation results demonstrate the effectiveness of our proposed schemes in achieving low average delay and high cache hit rate.

REFERENCES

[1] Ericsson. (2019). *Ericsson Mobility report*. [Online]. Available: https://www.ericsson.com/en/mobility-report

[2] K. Wang, F. R. Yu, H. Li, and Z. Li, "Information-centric wireless networks with virtualization and D2D communications," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 104–111, Jun. 2017.

[3] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[4] P. Ostovari, J. Wu, A. Khreishah, and N. B. Shroff, "Scalable video streaming with helper nodes using random linear network coding," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1574–1587, Jun. 2016.

[5] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[6] K. Poularakis and L. Tassiulas, "On the complexity of optimal content placement in hierarchical caching networks," *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 2092–2103, May 2016.

[7] K. Poularakis and L. Tassiulas, "Optimal cooperative content placement algorithms in hierarchical cache topologies," in *Proc. 46th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2012, pp. 1–6.

[8] T. X. Tran, D. V. Le, G. Yue, and D. Pompili, "Cooperative hierarchical caching and request scheduling in a cloud radio access network," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2729–2743, Dec. 2018.

[9] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized Quality-of-Experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[10] M. Choi, J. Kim, and J. Moon, "Wireless video caching and dynamic streaming under differentiated quality requirements," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1245–1257, Jun. 2018.

[11] X. Zhang, Y. Ren, H. Gao, T. Lv, and Y. Lu, "Analysis of caching and transmitting scalable videos in cache-enabled small cell networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[12] X. Zhang, T. Lv, W. Ni, J. M. Cioffi, N. C. Beaulieu, and Y. J. Guo, "Energy-efficient caching for scalable videos in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1802–1815, Aug. 2018.

[13] D. Zhu, H. Lu, Z. Gu, Y. Lu, and F. Guo, "Joint power allocation and caching for SVC videos in heterogeneous networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[14] X. Zhang, T. Lv, and S. Yang, "Near-optimal layer placement for scalable videos in cache-enabled small-cell networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 9047–9051, Sep. 2018.

[15] D. Jiang and Y. Cui, "Analysis and optimization of caching and multicasting for multi-quality videos in large-scale wireless networks," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 4913–4927, Jul. 2019.

[16] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, "Caching and operator cooperation policies for layered video content delivery," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[17] C. Zhan and Z. Wen, "Content cache placement for scalable video in heterogeneous wireless network," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2714–2717, Dec. 2017.

[18] B. Jedari and M. Di Francesco, "Delay analysis of layered video caching in crowdsourced heterogeneous wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[19] S. Ullah, K. Thar, and C. S. Hong, "Management of scalable video streaming in information centric networking," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21519–21546, Oct. 2017.

[20] Y. Wei, C. Xu, M. Wang, and J. Guan, "Cache management for adaptive scalable video streaming in vehicular content-centric network," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Jul. 2016, pp. 410–414.

[21] M. Xing, S. Xiang, and L. Cai, "A real-time adaptive algorithm for video streaming over multiple wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 795–805, Apr. 2014.

[22] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for PCS networks," in *Proc. 18th Annu. Joint Conf. Comput. Commun. Societies. Future Now*, 1999, pp. 1377–1384.

[23] D. Lopez-Perez, I. Guvenc, G. de la Roche, M. Kountouris, T. Quek, and J. Zhang, "Enhanced intercell interference coordination challenges in heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 22–30, Jun. 2011.

[24] Z. Zhang and D. Liu, "A distributed scheduling algorithm for heterogeneous cache-enabled small cell networks using ADMM," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, Sep. 2015, pp. 1–5.

[25] T. Han and N. Ansari, "A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1038–1051, Apr. 2016.

[26] L. Chen, Y. Zhou, and D. M. Chiu, "Smart streaming for online video services," *IEEE Trans. Multimedia*, vol. 17, no. 4, pp. 485–497, Apr. 2015.

[27] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. 7th ACM SIGCOMM Conf. Internet Meas. (IMC)*, 2007, pp. 1–14.

[28] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "LORM: Learning to optimize for resource management in wireless networks with few training samples," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 665–679, Jan. 2020.

[29] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 2019.

[30] W. Dan and H. Zhu, *Sublinear Algorithms for Big Data Applications*. Berlin, Germany: Springer, 2015.

[31] C.-B. Yu, J.-J. Hu, R. Li, S.-H. Deng, and R.-M. Yang, "Node fault diagnosis in WSN based on RS and SVM," in *Proc. Int. Conf. Wireless Commun. Sensor Netw.*, Dec. 2014, pp. 153–156.

[32] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[33] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[34] H. Kalva, V. Adzic, and B. Furht, "Comparing MPEG AVC and SVC for adaptive HTTP streaming," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2012, pp. 158–159.

ZHILONG ZHANG (Member, IEEE) received the B.E. degree in communication engineering from the University of Science and Technology, Beijing, China, in 2007, and the M.S. and Ph.D. degrees in communication and information systems from the Beijing University of Posts and Telecommunications (BUPT), Beijing, in 2010 and 2016, respectively. From 2010 to 2012, he was a Software Engineer with TD Tech Ltd., Beijing. From 2014 to 2015, he was a Visiting Scholar with Stony Brook University, Stony Brook, NY, USA. He is currently a Lecturer with BUPT. His research interests include optimization theory and its applications in wireless multimedia networks and mm-wave communications.

JIANMEI DAI (Member, IEEE) received the B.S. degree in communication engineering and the M.S. degree in communication and information systems, in 2004 and 2007, respectively. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications, Beijing. He was a Visiting Scholar with Auburn University, Auburn, AL, USA, in April 2019. He is currently a Lecturer with the Beijing University of Posts and Telecommunications. His research interests include optimization theory and its applications in wireless video transmission and wireless networks.

MINYIN ZENG received the B.E. degree in communication engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2018, where he is currently pursuing the M.S. degree. His research interests include wireless video transmission and wireless resource management.

DANPU LIU (Senior Member, IEEE) received the Ph.D. degree in communication and electrical systems from the Beijing University of Posts and Telecommunications, Beijing, China, in 1998. She was a Visiting Scholar with the City University of Hong Kong, in 2002, The University of Manchester, in 2005, and the Georgia Institute of Technology, in 2014. She is currently with the Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications. She has published over 100 articles and three teaching books. She holds over 26 patent applications. Her research involved with MIMO, OFDM, and broadband wireless access systems. Her recent research interests include 60GHz mm-wave communication, wireless high-definition video transmission, and wireless sensor networks.

SHIWEN MAO (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA. He is currently a Samuel Ginn Professor with the Department of Electrical and Computer Engineering and the Director of the Wireless Engineering Research and Education Center (WEREC) with Auburn University, Auburn, AL, USA. His research interests include wireless networks, multimedia communications, and smart grid. He received the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award, in 2019 and NSF CAREER Award, in 2010. He was a co-recipient of the IEEE ComSoc MMTC Best Conference Paper Award, in 2018, the Best Demo Award from the IEEE SECON, in 2017, the Best Paper Awards from the IEEE GLOBECOM 2019, in 2016 and 2015, the IEEE WCNC, in 2015, the IEEE ICC, in 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in communications systems. He serves the Editorial Board of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE INTERNET OF THINGS JOURNAL, the IEEE MULTIMEDIA, the IEEE NETWORKING LETTERS, and *ACM GetMobile*, and so on.