Utility Maximization of Cloud-based In-Car Video Recording over Vehicular Access Networks

Zhaobin Deng, Yipeng Zhou, Di Wu, Guoqiao Ye, Min Chen, Liang Xiao

Abstract

With the advance of cloud computing and 4G/5G technology, video contents recorded by in-car cameras (i.e., vehicular digital video recorders) can be uploaded to the cloud to facilitate accident analysis, online surveillance, video sharing, etc. However, the cost of uploading such huge volume of video contents via unstable vehicular access networks (including cellular base stations and roadside units (RSUs)) can be considerable by considering the increasing video quality requirement, time constraint and limited local buffer space. In this paper, we propose an adaptive video recording and uploading scheme to maximize the overall utility of cloud-based in-car video uploading over vehicular access networks. Specifically, the utility function is defined as the weighted sum of bandwidth cost and video quality and we formulate the problem into a constrained Markov Decision Process (MDP). Based on the theoretic foundation of MDP, we design and implement an algorithm to obtain an adaptive chunk uploading policy for video contents over vehicular access networks. Extensive simulations have been conducted to demonstrate that our policy can achieve the best performance compared with other alternative strategies.
I. INTRODUCTION

With the cost dropping sharply, in-car cameras (or vehicular digital video recorders (VDVRs)) have been widely installed on vehicles. As vehicles move along roads, VDVRs can record street views and events (e.g., accidents) on the road. Such video contents are essential for many applications such as accident analysis, online surveillance, video sharing and tamper detection (e.g., [1], [2], [3]). However, it is challenging to efficiently transmit the massive video content from vehicles to remote cloud servers via unstable vehicular access networks mainly constituted by cellular base stations and RSUs. Due to limited buffer space, if the video content cannot be uploaded to servers in time, stale video content will be overwritten by fresh video content in VDVRs, resulting in video interruptions. Unfortunately, the channel of vehicular access networks is very volatile when a vehicle is moving. It is desirable to upload video contents with adaptive bitrates according to the latest network conditions.

Dynamic Adaptive Streaming over HTTP (DASH) is a well-known technique developed to automatically tune the bitrate for video streaming according to the latest network conditions such that the frequency of streaming interruptions can be minimized. Since DASH is particularly effective when the network capacity is unstable and fluctuates over time, it has been widely used by real systems [4] [5] [6]. With similar principle, it is also feasible to adopt the idea of DASH to adapt bitrates for video uploading in the vehicular access networks. However, there are two obstacles that should be carefully considered: firstly, the capacity of vehicle access networks can be extremely volatile as vehicles are moving; secondly, the local VDVR’s buffer space is limited and it is essential to avoid overflowing the VDVR’s buffer even if failing to upload contents in time.

The scenario addressed by our work can be described as follows. First of all, the video content is continuously generated by a VDVR and the video content can be divided into discrete chunks with equal size. It is optional to transcode an original chunk into a new chunk with smaller size by lowering its bitrate. We need a strategy to choose an appropriate bitrate for each chunk to complete chunk transcoding before the chunk is uploaded to the cloud via cellular base stations.
or RSUs. Briefly speaking, we need to consider the following factors to determine a chunk’s bitrate: First, there is a deadline associated with each chunk since its creation before which it must be uploaded to the cloud, otherwise the chunk will be discarded to save space for fresher contents; second, the uploading rate cannot exceed the uploading capacity which is limited and can vary significantly as vehicles move; Third, the utilization of cellular networks should be minimized because it incurs additional monetary cost; fourth, it is desired to upload each chunk with the highest possible bitrate so as to maximize video quality. In summary, the design of an adaptive video uploading strategy is not a trivial problem complicated by these factors discussed above.

In this paper, to address the above problems, we formulate the adaptive video uploading process as a Markov decision process (MDP) with the objective to maximize utility (used exchangeably with time average reward which is a popular criterion to deal with problems formulated as infinite-horizon MDP [7]). In our work, utility is defined as a weighted sum of bandwidth cost and video quality. Then, the MDP problem can be tackled by analyzing the MDP with deviation matrix. In other words, we can approximately achieve the maximum utility (i.e., average time reward) as long as the chunk size is sufficiently small. Furthermore, based on analysis results, we design and implement an algorithm to obtain an adaptive video bitrate selection policy, which can gradually approach the optimal state after a few number of iterations. Our work is important for system designers to develop cloud-based in-car video recording systems, and evaluate the performance of similar systems. For users, this work can guide them to make a better tradeoff between bandwidth cost and video quality. For brevity, our contributions are summarized as follows:

- We are the first to propose an online policy to make decisions dynamically for adaptive cloud video uploading during driving periods with respect to the tolerable delay of a user.
- We formulate the problem into a constrained MDP problem with the objective to maximize utility, which well balances the tradeoff between QoE and monetary cost.
- We design an algorithm to obtain an adaptive bitrate selection policy by exploiting MDP theory, which takes current network condition, user QoE, bandwidth cost and tunable latency jointly into account. The algorithm can quickly converge to the optimal state.
- We conduct extensive simulations to evaluate the effectiveness of the proposed algorithm. The experimental results indicate that our policy can improve the overall utility (i.e., reward)
significantly in comparison with other alternative strategies.

The rest of this paper is organized as follows. We review previous works in Sec. II. Sec. III depicts the system model. In Sec. IV, we elaborate the problem formulation and Sec. V illustrates the analysis of the MDP and the design of our algorithm. The simulation settings and performance evaluation results are presented in Sec. VI. Finally, Sec. VII concludes the paper and points out our future work.

II. RELATED WORK

With the decrease of the cost for digital video recorders, VDVRs become affordable and have been widely installed on vehicles. A VDVR prototype system [8] can support real-time navigator, vehicle data encoding, secure data generation and offline video playing.

Pinilla et al. [1] proposed a diagnosis scheme based on fuzzy logic, which can diagnose people’s driving skills under real driving conditions using GPS data and video records. Kim et al. [3] proposed a new secure scheme by embedding watermarks in the I-frames to authenticate in-vehicle video, which can confirm the authentication and tamper detection in car surveillance videos. Lin et al. [9] proposed a speed warning system to predict the trend of changing speed by image analysis, feature detection, and Hidden Markov Model. Hsu et al. [10] exploited real-time video surveillance using VDVRs and proposed a web-based design by building peer-to-peer connections once a surveillance service is initiated. Chen et al. [2] proposed a speed-adaptive street view image generation system, which can efficiently generate street view images from massive video collections under different lightings and weather conditions. Abduljalil et al. [11] focused on storage saving using VDVRs and proposed a real-time framework, which enables users to get vehicle accident videos with a small amount of storage space. In our work, we focus on uploading video content generated by VDVRs through both vehicular access networks (including cellular and WiFi networks), in which the latency can be tunable according to users’ requirements.

In the field of traffic offloading, there are works of literature devoted in mobile offloading to reduce the exponentially increasing cellular traffic. Gao et al. [12] proposed a hybrid data dissemination model in Fog computing systems by utilizing both DTNs and Cloud techniques. Dimatteo et al. [13] proposed an architecture MADNet to offload cellular traffic by integrating RSUs and mobile-to-mobile communications. Intelligent Transportation System ITS is proposed
in [14] to enhance the offloading performance based on any prediction scheme. Salahuddin et al. [15] proposed an RSU cloud as the operational backbone of vehicle grid in IoV to efficiently serve the underlying demand from the vehicle grid. Some additional works [16]–[18] focused on RSUs deployment to enhance throughput or reduce overhead.

Lee et al. [19] studied the offloading and battery power saving efficiency of WiFi with respect to different delay. Zhu et al. [20] established a mathematical framework to study a mobile data offloading system which integrates cellular networks and vehicular opportunistic communications and derived an optimal scheme for mobile data offloading. To improve handover efficiency, Cia et al. [21] exploited mobility patterns between cell coverage areas and road traffic congestion levels to optimize the handover bias in HetNets and to reduce handover completion times. Petrov et al. [22] designed a novel analytical model that jointly takes the features of relay based crowd sensing, the mobility of assisting vehicles, and the important features of emerging NB-IoT technology into account, which reveals the effect of vehicle-based relay. [23]–[25] employed queuing theory to analyze the process of opportunistic WiFi offloading in mobile networks, but none of them gave out an online uploading/downloading content policy in term of users. Other techniques such as erasure coding [26] are also utilized to improve data dissemination efficiency in opportunistic vehicular networks. Although there were a number of previous studies about WiFi offloading along the driving tour, none of them considered the scenario of VDVR video uploading in which video contents are generated at a constant rate. Several works such as [24] [25] tried to optimize their objective just by tuning some static parameters. Our method is more advanced by making chunk uploading decisions adaptively.

In the field of video transmission over vehicular access networks, Vinel et al. [27] measured the quality of video transmitted over vehicle access networks and found that good visual quality of transmitted videos can be achieved when the distance between vehicles is less than 400 meters. Wu et al. [28] considered the interference of neighboring nodes for video forwarding, and proposed a routing algorithm that can improve the quality of end-to-end transmission video. Xie et al. [29] examined the impact of different buffer management schemes on the performance of video streaming between vehicles and proposed a buffer management scheme called EDF to improve video quality. Different from above work, we focus on designing a bitrate selection policy from the perspective of users.
III. System Model

To facilitate the understanding of the scenario studied by our work, we describe the video uploading process via vehicular access networks in Fig. 1. In the figure, as vehicles move on the road, they can either contact RSUs of WiFi networks or cellular networks. Vehicles can upload video content through cellular and WiFi networks to remote cloud. Cellular networks are always available for connection, however the WiFi networks are only intermittently available. Figure 1 shows a practical scenario with multiple vehicles in motion. Without loss of generality, our discussion focuses on the case with a single vehicle. In fact, it is not difficult to generalize our model to the case with multiple users by incorporating the distribution of the number of vehicles covered by each RSU because the number of competing vehicles accessing the same RSU can also affect available WiFi capacity.

Since cellular networks cost money, WiFi is always connected with higher priority. If the vehicles are out of the range covered by any RSU, they can either choose to upload content via cellular networks or suspend uploading until they reach the next RSU. Note that suspending video uploading is a risky option since VDVRs’ buffers could be overflowed given their limited space. Therefore, as vehicles move, they switch their video transmissions between uploading
and suspending from time to time. Therefore, our algorithm needs to not only select the right bitrate for each chunk but also decide the time to switch transmissions.

A. Notations

To smooth the presentation of our model, the mathematical notations used in our paper are summarized in Table I.

B. Video Transcoding

Video transcoding is the most important operation for us to adapt the video bitrate. The time is split into different time slots and the duration of each is $\tau$. In general, we can assume that the video content recorded by VDVRs in one time slot can be split into $k$ video chunks. The original bitrate of each chunk is denoted by $b_1$ implying that the size of each original chunk is $b_1 \cdot \tau / k$ and $k$ new chunks are generated at the end of each time slot.

A video chunk can be transcoded into another with the bitrate chosen by our algorithm before it is uploaded to the cloud. We let $B = \{b_1, b_2, \ldots, b_{|B|}\}$ with $b_1 > b_2 > \cdots > b_{|B|}$, where $|B|$ is the number of all possible bitrates. Note that transcoding operation can only lower bitrate, thus there is no bitrate exceeding the original bitrate $b_1$. Higher bitrate means better video quality but consumes more bandwidth resource.

C. Constraints

As we have discussed, the problem we confront has two kinds of constraints: limited buffer space and limited bandwidth resources. Our modeling should take these constraints into account.

1) Buffer Constraints: The maximum size of a VDVR’s buffer is denoted by $b_1 \cdot \tau / k \cdot G$, where $G$ represents the limited capacity of the storage, i.e., the maximal number of video chunks a VDVR can store. On the other hand, users can customize the maximal delay $D$, namely the maximal number of slots video chunks can stay in buffer at most for their personal requirements. If the VDVR does not upload content during the user delay tolerance period, there are at most $k \cdot D$ non-uploaded video chunks in the buffer. In other words, the maximal number of non-uploaded video chunks $M$ that the buffer can hold is determined by the buffer size $G$ and users’ tolerant delay $D$, which is equivalent to $M = \min(G, k \cdot D)$. Let $m(t)$ denote the number
TABLE I
KEY MATHEMATICAL NOTATIONS IN SYSTEM MODEL

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>The length of a time slot.</td>
</tr>
<tr>
<td>$k$</td>
<td>The number of chunks produced within a time slot.</td>
</tr>
<tr>
<td>$B$</td>
<td>The set of bitrate which is selected in VDVRs’ transcoding operation.</td>
</tr>
<tr>
<td>$C$</td>
<td>The set of bandwidth of cellular networks.</td>
</tr>
<tr>
<td>$W$</td>
<td>The set of bandwidth of WiFi networks.</td>
</tr>
<tr>
<td>$c(t)$</td>
<td>The bandwidth of cellular networks at time slot $t$.</td>
</tr>
<tr>
<td>$w(t)$</td>
<td>The bandwidth of WiFi networks at time slot $t$.</td>
</tr>
<tr>
<td>$m(t)$</td>
<td>The number of video chunks in buffer at time slot $t$.</td>
</tr>
<tr>
<td>$G$</td>
<td>The maximal number of video chunks limited by the capacity of VDVRs.</td>
</tr>
<tr>
<td>$D$</td>
<td>Tunable delay, denoted by the maximal number of time slots specified by users’ personal requirements.</td>
</tr>
<tr>
<td>$M$</td>
<td>The maximal number of non-uploaded video chunks that the VDVR can hold, determined by $G$ and $D$.</td>
</tr>
<tr>
<td>$n^c(t)$</td>
<td>The number of video chunks uploaded via cellular networks at time slot $t$.</td>
</tr>
<tr>
<td>$n^w(t)$</td>
<td>The number of video chunks uploaded via WiFi networks at time slot $t$.</td>
</tr>
<tr>
<td>$b^c(t)$</td>
<td>The bitrate of chunks uploaded via cellular networks at time slot $t$.</td>
</tr>
<tr>
<td>$b^w(t)$</td>
<td>The bitrate of chunks uploaded via WiFi networks at time slot $t$.</td>
</tr>
<tr>
<td>$1/\lambda_c$</td>
<td>The expected duration of WiFi-available period.</td>
</tr>
<tr>
<td>$1/\lambda_f$</td>
<td>The expected duration of WiFi-unavailable period.</td>
</tr>
<tr>
<td>$\sigma_c$</td>
<td>The scale parameter of cellular bandwidth distribution.</td>
</tr>
<tr>
<td>$\sigma_w$</td>
<td>The scale parameter of WiFi bandwidth distribution.</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>The tradeoff between QoE and bandwidth cost.</td>
</tr>
<tr>
<td>$Q(t)$</td>
<td>The QoE of users at time slot $t$.</td>
</tr>
<tr>
<td>$C(t)$</td>
<td>The cost of users at time slot $t$.</td>
</tr>
</tbody>
</table>
of video chunks in the buffer at time slot $t$. To avoid video interruptions and meet the delay constraint, the following inequation should be satisfied,

$$m(t) \leq M, \quad t = 0, 1, \ldots$$  \hspace{1cm} (1)

For simplicity, we have a hypothesis that the lowest transcoded bitrate $b_{\min}$ is less than the lowest capacity of cellular networks. Given this hypothesis, above inequation can always hold and buffer overflow can be avoided since network capacity is always sufficient to upload chunks with the lowest bitrate.

Let $n_c(t)$ and $n_w(t)$ denote the number of video chunks being uploaded to the cloud via cellular networks and WiFi networks respectively at time slot $t$. Then,

$$m(t) - (n_c(t) + n_w(t)) + k \leq M, \quad \forall t$$ \hspace{1cm} (a)

$$n_c(t) + n_w(t) \leq m(t), \quad \forall t.$$ \hspace{1cm} (b)

Constraint $(a)$ is used to ensure that buffer should not overflow at next time slot and video chunks must be uploaded within users’ tolerant delay. Constraint $(b)$ guarantees that the number of video chunks a VDVR can transmit is at most equal to the number of video chunks remaining in buffer.

2) **Bandwidth Constraints:** Bandwidth constraints mean that the uploading rate cannot exceed the network capacity at any moment. For simplicity, we assume that the bandwidth of cellular and WiFi networks is immutable within a time slot. This assumption approximately holds since the length of each time slot $\tau$ is quite short. We denote the bandwidth of cellular and WiFi networks at time slot $t$ by $c(t)$ and $w(t)$ respectively.

We divide the range of bandwidth of cellular networks into different values, and let $C = \{c_1, c_2, \ldots, c_{|C|}\}$ denote the available bandwidth set with $0 < c_1 < c_2 < \cdots < c_{|C|}$. The capacity of cellular networks must satisfy $c(t) \in C, \forall t \in N^*$. Without loss of generality, we assume that the bandwidth capacity of a cellular network follows the Rayleigh distribution which is widely used to simulate the wireless conditions as [30] with parameter $\sigma_c$ implying that the bandwidth capacity of a cellular network is an independent identically distributed (i.i.d.) variable at different time slots.

Similarly, the bandwidth of a WiFi network can only take different values from the set denoted by $W = \{w_0, w_1, w_2, \ldots, w_{|W|}\}$ with $0 = w_0 < w_1 < w_2 < \cdots < w_{|W|}$. Note that the bandwidth
capacity of a WiFi network can be 0 representing that the vehicle is outside the range of any RSU. We also take the bandwidth capacity of a WiFi network denoted by $w(t) \in \mathcal{W}$ where $w(t) \neq 0$ as an i.i.d. variable at different time slots following the Rayleigh distribution with parameter $\sigma_w$. Furthermore, the availability of WiFi networks can be modeled as an ON-OFF alternating renewal process. We assume that both ON period and OFF period are exponentially distributed with parameters $\lambda_e$ and $\lambda_f$ respectively. So the number of time slots consecutively available (unavailable) for WiFi networks approximately obeys exponential distribution with parameter $\lambda_e \tau (\lambda_f \tau)$.

It is feasible to track the location of moving vehicles. GPS is the most commonly used positioning technique to obtain geographical information. Network-based location technology [31] relies on the deployed cellular or WiFi networks to determine the position of a mobile user by measuring its signal parameters. Other localization technologies such as RFID-based TrackT [32] and TMicroscope [33] can also be used to achieve mobile node positioning. The location of vehicles and RSUs contains useful information which can help our model predict the bandwidth and the availability of WiFi networks more accurately. To verify this point, a set of simulations are conducted in Sec. VI-G.

Note that a video coding rate can be maintained by multiple transmission rates. For simplicity, we assume that video coding rate is equal to video transmission rate as the assumption made in [34] [35] [36]. We also assume that the bitrate of video chunks uploaded through the same way (Cellular or WiFi) in the same time slot are identical\(^1\). Then, the constraints with limited network capacity are:

\[ b^c(t) \cdot n^c(t)/k \leq c(t), \forall t \quad \text{(c)} \]
\[ b^w(t) \cdot n^w(t)/k \leq w(t), \forall t \quad \text{(d)} \]

Here $b^c(t)$ and $b^w(t)$ are the bitrates of chunks uploaded via cellular networks and WiFi networks respectively.

**Discussion:** Since the number of uploaded chunks must be an integer in each time slot, it is possible that the bandwidth resource cannot be fully utilized resulting in some bandwidth

\(^1\)Here, we do not allow chunks uploaded within the same time slot to have different bitrates so as to minimize bitrate variation. From user's perspective, video bitrate variation could be annoying. Video streaming with stable bitrate is more preferred according to QoE study [37].
wastage. This wastage becomes negligible if we enlarge the number of video chunks produced in a single time slot because wasted bandwidth in a single time slot is no more than the size of a single chunk $b_1 \cdot \tau/k$. To verify this point, a simulation is conducted in Sec. VI-B.

D. Utility

Our objective is to maximize utility, which is a metric defined as follows:

$$
\bar{R} = \lim_{T \to +\infty} \frac{1}{T+1} \sum_{t=0}^{T} (\alpha \cdot Q(t) - (1 - \alpha) \cdot C(t)).
$$

Here $Q(t)$ is the Quality-of-Experience (QoE) score and $C(t)$ is the bandwidth cost at time slot $t$ respectively, which are specified later. $\alpha$ is a tunable parameter through which users can balance video quality against bandwidth cost. It ranges from 0 (only bandwidth cost is considered in this extreme case) to 1 (only user QoE is considered in this extreme case). Utility is used exchangeably with time average reward, the term used as the metric for MDP analysis. Time average reward is commonly used as the objective for the problem formulated as an MDP. Average reward over time enables us to focus on the maximization of long-term gain instead of the performance in a short period through which we can optimize global performance.

1) QoE Model: We adopt the widely acknowledged measure of QoE to quantify video quality. According to the QoE defined for video streaming in [37], we define the quality of an uploaded video chunk with bitrate $b$ as

$$
QS(b) = \begin{cases} 
\frac{\tau}{k} a_1 \ln \frac{a_2 b}{b_1} & b_{|B|} \leq b \leq b_1, \\
0 & a_2 b < b_1,
\end{cases}
$$

where $a_1$ and $a_2$ are two positive constant parameters. Here the constant $\tau/k$ is to make the QoE score related to chunk size. In other words, a smaller chunk means lower QoE. Note that the QoE function which is differentiable and monotonically increasing in $[b_{|B|}, b_1]$ with upper bound $\frac{\tau}{k} a_1 \ln a_2$. $a_1$ and $a_2$ satisfy the constraint $a_1 \ln a_2 = 5$.

Recall that $b^{c}(t)$ and $b^{w}(t)$ are the bitrates of video chunks uploaded via cellular networks and via WiFi networks respectively at time slot $t$. Thus the video quality being uploaded at time slot $t$ can be represented as:

$$
Q(t) = n^{c}(t) \cdot QS(b^{c}(t)) + n^{w}(t) \cdot QS(b^{w}(t)).
$$
2) **Cost Model:** VDVRs can upload video chunks through two network channels: cellular networks and WiFi networks. But users have to pay money for the usage of cellular networks. In comparison with cellular networks, the cost of public WiFi networks is much lower. Thus in our model, we simply consider the bandwidth cost incurred by cellular traffic. Let $\omega$ denote the unit cost of cellular traffic. The monetary cost of transmission for a chunk with bitrate $b$ is:

$$\text{Cost}(b) = \omega \cdot b \cdot \tau / k.$$  

With preceding definitions, the cost a user suffers in time slot $t$ can be represented as:

$$C(t) = n^c(t) \cdot \text{Cost}(b^c(t)).$$

Briefly speaking, the utility defined in Eq. (2) contains two parts: the first part is the long-term time-average QoE, and the second part is the minus average cost incurred by cellular bandwidth consumption. Therefore, we should try to maximize utility defined in Eq. (2).

Above discussion is used to illustrate the problem studied by us. We implicitly assume that the VDVRs are always working for video recoding such that our objective is to maximize the reward by letting time approach infinity. Since next section, we begin to introduce how to create the MDP problem based on the constraints discussed in this section.

### IV. Modeling

In this section, we first define the state space, and then specify the action space for each state, based on which we further define the transition matrix by calculating the transition probability between any two states. With all these fundamental definitions, we create the Markov decision process.

#### A. State Space $S$

State space is the set of all possible states the system could be. Note that the state of the system is the basis for us to make the chunk uploading decisions. Thus, a state of the system can be determined by the resource state, i.e., cellular network bandwidth, WiFi network bandwidth and used capacity of VDVRs’ buffer. Thus we define state space as $S = \{s = (c, w, m) \mid \forall c \in C, \forall w \in W, \forall m \in M\}$, where $c$ and $w$ are the bandwidths of cellular and WiFi networks of current state respectively. $M$ is a set of values which contains all the available numbers of not uploaded video chunks in buffer and it can be represented as $M = \{m \mid m \in N^*, m \leq M\}$. Obviously, for any time slot, the state of the system is unique.
B. Action Space $A$

An action is defined by a tuple with four elements, i.e., $a = (n^c, n^w, b^c, b^w)$, which determines both the number and bitrate of chunks to be uploaded via cellular networks and WiFi. For any given state $s$, the action space should contain all feasible actions we can make. In other words, a feasible action should not break the restrictions of limited bandwidth and buffer space. We define an indicator function $L(a, s)$ that can judge whether an action is feasible or not for a given state $s$.

$$L(a, s) = \begin{cases} 
1 & n^c, n^w \in \mathcal{M}, \\
& b^c, b^w \in \mathcal{B}, \\
& n^c \cdot b^c / k \leq c, \\
& n^w \cdot b^w / k \leq w, \\
& m - (n^c + n^w) + k \leq M, \\
& n^c + n^w \leq m, \\
0 & \text{else}.
\end{cases} \quad (7)$$

The first and second constraints in the function of $L$ returning 1 guarantee the number of chunks and their bitrates should be taken from $\mathcal{M}$ and $\mathcal{B}$ respectively. The third and fourth constraints guarantee that the throughput resulted by action $a$ cannot exceed the capacity of current network condition given current state $s$. The fifth constraint is used to avoid buffer overflow and no chunk will miss its upload deadline. The last constraint ensures that the total number of chunks to be uploaded is no more than the number of available chunks in the buffer.

We define an action space for a state $s \in \mathcal{S}$ as $\mathcal{A}(s) = \{ a = (n^c, n^w, b^c, b^w) \mid L(a, s) = 1 \}$, where $n^c, n^w, b^c$ and $b^w$ represent the numbers of uploaded video chunks and the bitrates of uploaded video chunks via cellular networks and WiFi respectively.

C. Policy

For the MDP, we need a policy to make chunk uploading decisions for each time slot. Thus, we define a policy $\psi$ as an infinite sequence that tells us an action on each time slot, i.e., $\psi = (\pi_1(s(1)), \pi_2(s(2)), \ldots, \pi_t(s(t)), \ldots)$. Here, $\pi_t(s(t))$ is a stochastic variable agreeing a probability distribution over space $\mathcal{A}(s(t))$, representing each valid action to be chosen with probability given state $s(t)$ at time slot $t$. 
Given the definition of the policy, the search space of the policy denoted by \( \Pi \) is infinite. To simplify the problem, we narrow down the search space by defining decision function \( f \) as follow. Decision function \( f \) can return a fixed valid action \( a \) for a given state \( s \). The search space of the decision function is denoted by \( \mathcal{F} \), which is finite since both state space and action space are finite. Later on, we will prove that the \( \bar{R} \) of the optimal decision function \( f^* \) is no less than that achieved by the optimal policy \( \psi^* \).

D. Transition Probability

In MDP, the system state will evolve with time. The change of network conditions and the chunk upload decision jointly determine how the system state changes between time slots. We specify the formula of \( P_f(s' \mid s) \) describing the probability to change from state \( s' \) to state \( s \) using decision function \( f \). In particular, \( s = (c, w, m) \) and \( s' = (c', w', m') \). Then,

\[
P_f(s \mid s') = P^c(c \mid c') \cdot P^W(w \mid w') \cdot P^M_f(m \mid m'),
\]

where \( P^c(c \mid c') \), \( P^W(w \mid w') \) and \( P^M_f(m \mid m') \) are the probabilities to change from \( c' \) to \( c \) for cellular networks, from \( w' \) to \( w \) for WiFi and from \( m' \) to \( m \) for buffer state respectively.

Note that \( P^c(c \mid c') \) and \( P^W(w \mid w') \) do not depend on the action made by the decision function \( f \). In theory, we can apply any models created for the changing of network capacity for cellular networks and WiFi. In this paper, we adopt a most commonly used model by assuming that the bandwidth of WiFi networks obeys Rayleigh distribution as mentioned above when WiFi is available. The last times of On and OFF (corresponding to availability and unavailability of WiFi) periods of WiFi obey exponential distribution with parameters \( \lambda_e \) and \( \lambda_f \) respectively. So,

\[
P^W(w_i \mid w') = \begin{cases} 
1 - e^{-\lambda_e \tau} & i = 0, w' > 0, \\
e^{-\lambda_e \tau} \int_{w_i}^{w_{i-1}} \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} dx & 0 < i < |\mathcal{W}|, w' > 0, \\
e^{-\lambda_e \tau} \int_{w_i-1}^\infty \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} dx & i = |\mathcal{W}|, w' > 0, \\
e^{-\lambda_f \tau} & i = 0, w' = 0, \\
(1 - e^{-\lambda_f \tau}) \int_{w_i}^{w_{i-1}} \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} dx & 0 < i < |\mathcal{W}|, w' = 0, \\
(1 - e^{-\lambda_f \tau}) \int_{w_i-1}^\infty \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} dx & i = |\mathcal{W}|, w' = 0.
\end{cases}
\]
Similarly, we have

\[ P^c(c_i \mid c') = \begin{cases} 
\int_0^{c_i} \frac{x}{\sigma^2} e^{-\frac{x^2}{2 \sigma^2}} \, dx & i = 1, \\
\int_{c_{i-1}}^{c_i} \frac{x}{\sigma^2} e^{-\frac{x^2}{2 \sigma^2}} \, dx & 1 < i < |\mathcal{C}|, \\
\int_{c_{i-1}}^{\infty} \frac{x}{\sigma^2} e^{-\frac{x^2}{2 \sigma^2}} \, dx & i = |\mathcal{C}|. 
\end{cases} \] (10)

Note that the bandwidth state for both cellular and WiFi is memoryless.

In contrast, the change of buffer state depends on the action \( a \) returned by decision function \( f \). Let \( f(s') = (n^c, n^w, b^c, b^w) \) denote the action returned by \( f \), and we can easily obtain

\[ P^M_f(m \mid m') = \begin{cases} 
1 & m = m' - n^c - n^w + k, \\
0 & \text{else.} 
\end{cases} \] (11)

With all above discussion, for any given action \( a \), we can compute how the system state evolves from \( s' \) to \( s \), which is the basis for problem formulation presented in the next subsection.

**E. Formulation**

By introducing the concepts of state space, action space and decision function, we redefine the reward and formulate the problem as a stochastic Markov optimization problem.

For a given action \( a = (n^c, n^w, b^c, b^w) \), we immediately get the reward as:

\[ R(a) = \alpha \cdot (n^c \cdot QS(b^c) + n^w \cdot QS(b^w)) - (1-\alpha) \cdot n^c \cdot Cost(b^c). \] (12)

\( \alpha \) is the tradeoff parameter between QoE and bandwidth cost as mentioned above. We use notation \( g_{f,T} \) to denote the total reward over time slot 0 to time slot \( T \) by adopting decision function \( f \) with initial state \( s_i \) (i.e. \( s(0) = s_i \)), namely

\[ g_{f,T}(s_i) = \sum_{t=0}^{T} R(f(s(t))), i = 1, 2, \ldots, |S|, \] (13)

where \( s(0) = s_i \) and \(|S|\) is the total number of possible initial states. Note that the total reward \( g_{f,T}(s_i) \) is related to both decision function \( f \) and the initial state \( s_i \). The objective is to maximize all \( g_{f,T}(s_i), \forall s_i \) with different initial states. In fact, the decision function \( f^* \) optimizing the total reward with \( s(0) = s_i \) also optimizes the total reward with \( s(0) = s_j \). This point will be clear in the next section.
Let vector \( g_f,T = (g_f,T(s_1), g_f,T(s_2), \ldots, g_f,T(s|S|))^T \) denote the total reward with different initial states using decision function \( f \). We denote state transition matrix of decision function \( f \) as \( P_f \) where the transition probability between two states depends on the decision function \( f \). The element at \( i^{th} \) row and \( j^{th} \) column is:

\[
P_f(i,j) = P_f(s_j \mid s_i).
\] (14)

Similarly, we use a vector to express the reward at each state using a particular decision function \( f \), i.e., \( R_f = (R(f(s_1)), \ldots, R(f(s|S|)))^T \) a vector of which the \( i^{th} \) element is \( R(f(s_i)) \) representing the reward at state \( s_i \) using the decision function \( f \).

Now, we can concisely represent the vector of total expected reward as:

\[
g_f,T = \sum_{t=0}^{T} P_f^t R_f.
\] (15)

Here, \( P_f^t \) is the \( t \) power of the transition matrix \( P_f \). Thus, \( P_f^t R_f \) is the reward vector at \( t^{th} \) time slot. By letting \( T \) approach infinity, the problem can be formulated as:

\[
P1. \max_{\forall f \in F} g_f = \lim_{T \to \infty} \frac{g_f,T}{T+1}
\] (16)

s.t. \( f(s) \in A(s), \forall s \in S \).

Here, the objective \( g_f \) is the vector representing the average time reward with different initial states and its \( i^{th} \) element is denoted by \( g_f(s_i) \).

Recall that our original objective is the time average reward \( \bar{R} \) defined in the last section. In fact, \( \bar{R} \) can be considered as \( q \cdot g_f \), where \( q \) is the vector representing the probability distribution of initial states, i.e., \( q_i = P_r(s(0) = s_i) \). The only difference is that we can maximize each element in \( g_f \) so as to maximize the time average reward.

V. ALGORITHM

A. Analysis of MDP

Given the above expression of \( P1 \), we define its Cesaro limit as \( \lim_{T \to \infty} \frac{1}{T+1} \sum_{t=0}^{T} P_f^t \).

**Lemma 1** If \( P_f \) is the transition matrix with respect to decision function \( f \), the Cesaro limit of \( P_f \) would converge to a matrix denoted by \( P_f^* \), namely

\[
P_f^* = \lim_{T \to \infty} \frac{1}{T+1} \sum_{t=0}^{T} P_f^t.
\] (17)
And it satisfies:
\[ P_f^*P_f = P_fP_f^* = P_f^*P_f = P_f^*. \]  
(18)

Proof: Please refer to [38] for the proof details.

With Lemma 1, the existence of the Cesaro limit of \( P_f \) can be guaranteed. The calculation of the Cesaro limit \( P_f^* \) can be found in [39] which can be summarized as the calculation of steady-state distribution of each recurrent states. Thus the expected average reward can be expressed as:

\[ g_f = \lim_{T \to \infty} \frac{1}{T+1} \sum_{t=0}^{T} P_{tf}R_f = P_f^*R_f. \]  
(19)

Thus \( P1 \) can be transformed into \( P2 \):

\[ P2. \max_{f \in F} g_f = P_f^*R_f \]  
(20)

s.t. \( f(s) \in A(s), \forall s \in S. \)

We introduce the definitions of deviation matrix \( H_f \) and bias \( h_f \), which are used to find the optimal decision function \( f^* \). Their expressions are:

\[ H_f = (I - P_f + P_f^*)^{-1} - P_f^* \]  
(21)

and

\[ h_f = H_fR_f, \]  
(22)

where \( I \) is an identity matrix. The bias \( h_f \) represents the expected total difference between the reward and the stationary reward.

In order to solve problem \( P2 \), we first derive two inequations which can be leveraged to design the iterative algorithm to solve \( P2 \).

Lemma 2 For two decision functions \( f \) and \( f' \), if

\[ (I - P_{f'})g_f < 0 \]  
(23)

or

\[ (I - P_{f'})g_f = 0 \]  
(24)

\[ g_f - R_{f'} + (I - P_{f'})h_f < 0 \]  
(25)

then \( g_{f'} \geq g_f \).
Proof: The proof of this lemma is quite complicated. Please refer to the appendix for details.

From Lemma 2, we can improve the expected average reward with respect to any decision function \( f \) by seeking a new one satisfying Eq. (23) or Eq. (24) and Eq. (25).

With a little bit abusing of notations, we let \( P_a(s|s') \) represent the probability the state switch from \( s' \) to state \( s \) by adopting action \( a \). It is not difficult to derive the expression of \( P_a(s|s') \) based on the definition of \( P_f(s|s') \), which is omitted for brevity. In fact, \( P_a(s|s') = P_f(s|s') \) if \( f(s') = a \). We define two action sets for a decision function \( f \) with state \( s \):

\[
\mathcal{T}_1(s, f) = \{ a \in \mathcal{A}(s) \mid g_f(s) < \sum_{i=1}^{|S|} P_a(s_i|s) g_f(s_i) \}
\]
and

\[
\mathcal{T}_2(s, f) = \{ a \in \mathcal{A}(s) \mid \sum_{i=1}^{|S|} P_a(s_i|s) g_f(s_i) = g_f(s), \ h_f(s) + g_f(s) < R(a) + \sum_{i=1}^{|S|} P_a(s_i|s) h_f(s_i) \},
\]

where \( g_f(s) \) and \( h_f(s) \) are elements of vectors \( g_f \) and \( h_f \). We should note that actions in \( \mathcal{T}_1(s, f) \) and set \( \mathcal{T}_2(s, f) \) will be better than the action made by the current decision function \( f \) according to Lemma 2. Based on \( \mathcal{T}_1(s, f) \) and \( \mathcal{T}_2(s, f) \), we can iteratively improve decision function \( f \) with a given state \( s \) according to the following rules:

\[
f(s) = \begin{cases} 
  a_1 & \mathcal{T}_1(s, f) \neq \emptyset, \\
  a_1 & \in \arg\max_{a \in \mathcal{T}_1(s, f)} \sum_{i=1}^{|S|} P_a(s_i|s) g_f(s_i), \\\n  a_2 & \mathcal{T}_1(s, f) = \emptyset, \mathcal{T}_2(s, f) \neq \emptyset, \\
  a_2 & \in \arg\max_{a \in \mathcal{T}_2(s, f)} \{ R(a) + \sum_{i=1}^{|S|} P_a(s_i|s) h_f(s_i) \}, \\
  f(s) & \mathcal{T}_1(s, f) = \emptyset, \mathcal{T}_2(s, f) = \emptyset.
\end{cases}
\]

With rules defined in Eq. (28), we can iteratively update the decision function until \( \mathcal{T}_1(s, f) \) and \( \mathcal{T}_2(s, f) \) are empty for all states. Because action space \( \mathcal{A} \) is limited and the reward of decision function \( f \) will be monotonically lifted, \( \mathcal{T}_1(s, f) \) and \( \mathcal{T}_2(s, f) \) will eventually become empty.
We prove that the final decision function searched with Eq. (28) is an optimal decision function in space $\mathcal{F}$ achieving the maximum utility (i.e., reward defined in $\mathbf{P2}$).

**Theorem 1** Supposing a decision function $f_*$ which satisfies $\forall s \in \mathbf{S}$ with $\mathcal{T}_1(s, f_*) = \emptyset$ and $\mathcal{T}_2(s, f_*) = \emptyset$, $f_*$ is an optimal decision function for problem $\mathbf{P2}$.

**Proof:** Please read appendix for the proof details.

We further prove the optimality of $f_*$ in the whole policy space $\Pi$ in the following theorem.

**Theorem 2** For the problem defined in $\mathbf{P2}$, the utility (i.e., average reward) achieved by the optimal decision function $f_* \in \mathcal{F}$ and the maximum utility achieved by any policy $\psi_* \in \Pi$ are the same.

**Proof:** Please read appendix for the proof details.

### B. Algorithm Design

Based on above analysis, we design and implement an iterative algorithm to obtain the policy called VVUOP (VDVR Video Upload Online Policy) which can decide chunks’ bitrates and the number of chunks for uploading. The details are given in **Algorithm 1**.

Firstly, we initialize state space $\mathbf{S}$ and action space $\mathcal{A}(s)$ for each $s \in \mathbf{S}$ according to the parameters from environment and users’ preference such as limited capacity of VDVRs and tunable delay and so on. For each new iteration, the algorithm will produce a new strictly improved decision function based on the previous one. When the new decision function and the previous one are the same, the algorithm stops and we obtain the optimal decision function $f_*$. With limited state space and action space, the algorithm will terminate with finite steps. The detail of an iteration is presented as below.

i) We initialize the transition matrix $\mathbf{P}_f$ of current decision function $f$ by Eq. (14), and calculate its Cesaro limit which can be used to calculate the performance metric: expected average reward $g_f$ and bias $h_f$ of current decision function.

ii) We find action sets $\mathcal{T}_1(s, f)$ and $\mathcal{T}_2(s, f)$ by Eq. (26) and Eq. (27) for each $s \in \mathbf{S}$ with respect to current decision function. Based on Lemma 2, actions in $\mathcal{T}_1$ and $\mathcal{T}_2$ will induce a function with better average reward in comparison with the average reward of the current one.
**Algorithm 1** Algorithm of obtaining VVUOP

**Input:**

- Limited capacity of VDVR: $G$
- Tunable delay: $D$
- Sets of cellular and WiFi networks bandwidth: $C$, $W$
- Set of bitrate: $B$

**Output:**

Optimal decision function $f_*$

1. Initialize state space $S$ and action space $A(s)$ for each $s \in S$;
2. Let $f'$ be an arbitrary decision function;
3. repeat
   4. let $f = f'$;
   5. initialize the transition matrix $P_f$, and calculate the corresponding Cesaro limit $P_f^*$;
   6. calculate $g_f$ and $h_f$ by Eq. (19) and Eq. (22);
   7. calculate $T_1$ and $T_2$ for each state $s \in S$ with $g_f$ and $h_f$ obtained from step 6;
   8. update decision function $f$ according to Eq. (28) to obtain improved decision function $f'$;
   9. until $f' = f$
10. return $f_* = f$

iii) A new decision function $f'$ is obtained by replacing the decision actions with the actions in $T_1(s, f)$ and $T_2(s, f)$ yielding a function with better average reward.

The overall time complexity of **Algorithm 1** is $O(max\{|A||S|^2, |S|^{2.376}\})$, where $A = \cup_{s \in S} A(s)$ and $S$ is the state space. It is analyzed as follows. In our algorithm, the calculation of $P_f^*$ (line 5) is equivalent to solving linear equations [39] which can be transformed into a matrix-inversion problem. Thus its time complexity is $O(|S|^{2.376})$ according to the complexity of Coppersmith-Winograd algorithm [40]. The computations of $g_f$ and $h_f$ (line 6) comprise of matrix multiplication and matrix inversion as shown in Eq. (19) and Eq. (22) and the time complexities for solving both of them are $O(|S|^{2.376})$. For line 7 in **Algorithm 1**, the time complexity for calculating $T_1(s, f)$ and $T_2(s, f)$ is $O(|A(s)||S|)$ for a given state. Because we need to solve $T_1(s, f)$ and $T_2(s, f)$ for each state in $S$, the time complexity to solve $T_1$ and $T_2$
(line 7) is $O(\|A\|\|S\|^2)$ as an upper bound where $A = \bigcup_{s \in S} A(s)$. Thus, the time complexity of an iteration to update the decision function is $O(\max\{\|A\|\|S\|^2, \|S\|^{2.376}\})$ in total which is dependent on the scale of total action space $A$. According to our experiments, the number of iterations in our algorithm is quite small (no more than 10 in most cases).

It is common that there are thousands of states in $S$, and the complexity is acceptable given that the algorithm is executed in an offline manner. In other words, with the knowledge of the distributions for network conditions, the uploading policy can be fixed in advance. In addition, the running time of the algorithm can be shortened by distributing the computing tasks to multiple computers. There exist a number of studies [41] [42] [43] [44] [45] solving matrix operations by splitting the task into multiple subtasks so that they can be executed in parallel.

Finally, the contributions of Algorithm 1 are summarized as:

- **Algorithm 1** is the first one designed by modeling the video uploading problem with MDP. It can yield a policy that can adaptively adjust the chunks to be uploaded according to the latest network conditions.
- **Algorithm 1** can find the optimal policy achieving the best time average reward with polynomial time. It can be implemented as an offline algorithm and executed in parallel.

VI. PERFORMANCE EVALUATION

To demonstrate the superiority of our algorithm, we develop a simulator to evaluate the performance of our policy in comparison with other baseline algorithms. At first, we will describe our simulator settings before we present figures to compare algorithm performance.

A. Simulator Setting

<table>
<thead>
<tr>
<th>Sets</th>
<th>Elements (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cellular Set $C$</td>
<td>200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000</td>
</tr>
<tr>
<td>WiFi Set $W$</td>
<td>0, 800, 1600, 2000, 2500, 3000, 4000, 6000</td>
</tr>
<tr>
<td>Bitrate Set $B$</td>
<td>200, 400, 800, 1200, 1500, 2000</td>
</tr>
</tbody>
</table>
In our simulator, we imitate the vehicular network environment by computers. We set 10 bandwidth states for cellular networks, 8 bandwidth states for WiFi networks and 6 states for bitrates of video content, which are detailed in Table II. We set $\sigma_c = \sqrt{2 \cdot \frac{1000^2}{\pi}}$ implying that the mean bandwidth of cellular networks is 1000 Kbps according to the results in [46]. Fig. 2(a) is an example showing how the bandwidth of cellular networks fluctuates over 200 time slots following the Rayleigh distribution. The mean capacity of WiFi networks is set as 2700 Kbps, hence $\sigma_w = \sqrt{2 \cdot \frac{2700^2}{\pi}}$. This mean value is much larger than the average capacity of cellular
networks, which is consistent with the real world. Note that, WiFi bandwidth is 0 Kbps if no RSU is available. According to the reference [47], we set the price of cellular networks as $\omega = 0.1 \text{ yuan/Kb}$. It is reported that the average WiFi availability across cities is 11% [48]. Since we assume that WiFi availability obeys exponential distribution, we set $\lambda_e = \frac{1}{10}$ with mean duration equal to 10 seconds. Equivalently, the mean period without WiFi networks is 81 seconds. Here we define the WiFi availability ratio as $AR = \frac{E(WiFi-ON)}{E(WiFi-ON) + E(WiFi-OFF))}$ [25].

According to the video quality report [49], the original video content generated by VDVRs is 480p with bitrate equal to 2000 Kbps. VDVRs can transcode the video into 5 other versions with lower bitrate ranging from 200 Kbps to 2000 Kbps. The duration of each time slot $\tau$ lasts 2 seconds. For simplicity, we set $k = 1$ (i.e., the number of video chunks produced by VDVRs within a time slot is 1).

Recall that we use the function $QS(b)$ to represent user QoE where $b$ is the video bitrate [37]. For simulation, we set $a_2 = \frac{b}{|B|}$ and $a_1 = \frac{5}{\log(a_2)}$. With such setting, the QoE score will be 0 by uploading a chunk with the lowest bitrate. The maximum QoE score is 10 by uploading an original chunk.

To evaluate the performance of our VVUOP policy, we compare it with three candidates. The first two policies are canonical greedy strategies. The third one is a modified heuristic strategy designed in [50] which is originally adopted for DASH.

i) **Maximizing WiFi Utilization Upload (MWUU)** is a greedy policy that always maximizes the bitrate first for chunks uploaded via WiFi. Then, with fixed bitrate, MWUU decides the number of chunks for uploading. If WiFi is unavailable and the buffer is about to be overflowed, chunks will be uploaded via cellular networks with the lowest bitrate. This policy tries to minimize the use of cellular networks and exhaustively leverages WiFi networks to upload high-bitrate chunks with best effort.

ii) **Maximizing Reward Upload (MRU)** is also a greedy policy that always makes uploading decisions to maximize the reward for the current time slot.

iii) **Modified MPEG-DASH based Upload (MMDU)** is modified based on a strategy originally designed for DASH. Briefly speaking, the bitrate of the chunks uploaded at time slot $t$ is

---

2It is about 0.016 USD/kb.
decided by the buffer level. The formula is presented as below:

\[ b(t) = b_i, (i - 1)/|\mathcal{B}| \leq m(t)/M < i/|\mathcal{B}|. \] (29)

Here, \( M \) is the maximum buffer level and \( m(t) \) is the current buffer level. The bitrates for chunks uploaded for both WiFi and cellular networks are chosen according to this formula. Both cellular and WiFi networks will be used if both channel bandwidths are sufficient to support the selected bitrate. However, if the network capacity cannot support the selected bitrate, the uploading at this time slot will fail and no chunk will be uploaded.

For all our simulations, we use GSL (a scientific computing library) to simulate a scenario that a user drove (3600 slots) with the network conditions described above. To exclude randomness and present stable performance, we simulate 500 times to obtain reliable average results for each figure.

**B. Inaccuracy Evaluation**

Before we show simulation results, we would like to study how our assumptions can impact accuracy. Since we assumed that the number of uploaded chunks in each time slot must be an integer, a small amount of bandwidth will be wasted. To visualize how serious the wasted bandwidth is with different \( k \), we have conducted simulations of policy \( MWUU \) by setting \( k = 1, k = 3 \) and \( k = 5 \) respectively. Fig. 3 shows the CDFs for the ratio of wasted traffic to total

![CDF](image-url)
available throughput of WiFi networks. Apparently, we can note that the wasted traffic ratio declines as $k$ increases. The mean wasted traffic ratio is 5.27% with $k = 3$ and 1.36% with $k = 5$. Even for the worst case with $k = 1$, the mean of wasted traffic ratio is only 6.81%, an acceptable value in our simulations.

C. Comparison of average reward

![Graphs showing the comparison of average reward for different strategies.](image)

(a) Objective tradeoff parameter $\alpha = 0.1$. (b) Objective tradeoff parameter $\alpha = 0.5$. (c) Objective tradeoff parameter $\alpha = 0.9$.

Fig. 4. Comparison of average reward for different strategies by setting $\alpha = 0.1, 0.5, 0.9$ (i.e., objective tradeoff parameter) respectively.

Figure 4 depicts the average reward of all four policies with different objective tradeoff parameters. $\alpha = 0.1$ implies that users strongly care about their monetary cost. $\alpha = 0.5$ implies that QoE and cost are about equal important for users, and $\alpha = 0.9$ means that QoE dominates the objective function of Eq. (12). Here, we set the maximum tolerant delay as $D = 100$ smaller than the buffer capacity of VDVRs. This means a video chunk can stay in the buffer for at most 100 time slots. The results of this experiment show that our policy $VVUOP$ can achieve the highest average reward among all evaluated algorithms for all three cases with different $\alpha$. 
In Fig. 4(a), we can also note that the baseline algorithms cannot perform very well for all three cases. When $\alpha = 0.1$, $MWUU$ achieves the second highest average reward since users are very sensitive to their monetary payment and $MWUU$ tends to utilize WiFi networks as much as possible resulting in low cellular traffic cost. In contrast, the performance of $MMDU$ is much lower because the policy will leverage WiFi and cellular networks simultaneously without considering the cost of cellular networks. However, it performs quite well for cases with $\alpha = 0.5$ and $\alpha = 0.9$ due to the effective utilization of cellular networks. In the cases with $\alpha = 0.5$ and $\alpha = 0.9$, $MWUU$ performs the worst because users tend to be insensitive to monetary payment. $MWUU$ has the lowest priority to utilize cellular networks, and hence loses the bonus to improve average reward via cellular networks. $MRU$ tries to maximize the reward at each time slot. However, it fails to appropriate cellular networks in the case with $\alpha = 0.1$ because uploading chunks via cellular will degrade the reward in a single time slot. When QoE is weighted heavier for the cases with $\alpha = 0.5$ and $\alpha = 0.9$, $MRU$ tends to upload high-bitrate chunks so as to lift reward and in these cases cellular networks are utilized. However, the performance of $MRU$ is still strictly worse than $VVUOP$ indicating the superiority of our policy.

D. Average reward with different buffer limits

We evaluate the performance of four policies with different values of $M$, the capacity of VDVRs. $M$ is constant during the process of video transmission as assumed in [24] [29]. We examine the average reward under different values of $M$ and $\alpha$. The objective tradeoff parameter $\alpha$ is set to 0.1, 0.5 and 0.9. The simulation results are plotted in Fig. 5 showing that our policy always achieves the best performance with the highest average reward. It is also surprising to find that average rewards achieved by the baseline algorithms are very low in the first case with $M = 40$. The underlying reason is that $MWUU$, $MRU$ and $MMDU$ do not properly take occupied buffer length into consideration such that their average rewards are lower. Especially, the performance of baseline algorithms will be deteriorated more significantly with scant buffer capacity. In contrast, our policy $VVUOP$ can make use of the buffer capacity very well in all cases to achieve the best performance, and hence the performance gap will be largest in the first case with $M = 40$. For the other two cases, with plenty buffer capacity, the improvement of $VVUOP$ in comparison with $MRU$ and $MMDU$ is not very significant because buffer capacity is not the major bottleneck to affect average reward anymore. Therefore, it is crucial and meaningful to
Fig. 5. Comparison of average reward for different strategies with $M = 40, 70, 100$ and $\alpha = 0.1, 0.5, 0.9$, where $M$ is the buffer capacity and $\alpha$ is the objective tradeoff parameter.

**E. Average reward with different WiFi availability ratios**

AR (representing the probability of WiFi accessibility) is the ratio of the mean duration of WiFi period to the mean duration of WiFi-On and WiFi-Off period. Thus, AR represents the density of WiFi access points. We set AR equal to 0.11, 0.5 and 0.9 respectively, representing low, middle and high accessibility of WiFi networks. Fig. 2(b) shows the fluctuation of WiFi bandwidth in 200 time slots. From the figure, we can find that the greater the AR is, the greater the availability of the WiFi network is. To exclude the influence of buffer capacity on performance, we set $M$ to 70. We compare the performance of four policies in term of average reward with different ARs in Fig. 6.

From this figure, we note that with higher AR ratio, all policies can attain better performance for all cases with different objective tradeoff parameters ($\alpha$). It is straightforward that more traffic can be offloaded with denser RSUs providing WiFi access. Again, VVUOP is the best
Fig. 6. Comparison of average reward for different strategies with AR = 0.11, 0.5, 0.9 and α = 0.1, 0.5, 0.9, where AR is the WiFi accessibility ratio and α is the objective tradeoff parameter.

one achieving the highest average rewards for all cases. In other words, VVUOP can adjust the chunk uploading strategy according to the network conditions to gain highest utility. For other algorithms, we note that MWUU can only achieve good performance with high AR because MWUU is a conservative policy to minimize the use of cellular networks. MMDU selects bitrates for video chunks without considering the penalty from cellular traffic. Therefore, it can only achieve good performance for the case with a high α towards heavier weight on QoE. At last, MRU can achieve reasonable performances in all conditions because its decision making is reward oriented. However, MRU always achieves lower average rewards in comparison to VVUOP, because it is a greedy algorithm that only maximizes the reward at a single time slot without a global optimization solution.

F. Impact of objective tradeoff parameter

Recall that in the expression of our objective function, there is a tunable parameter α to balance between video quality and bandwidth cost. α is the weight of video quality and 1 − α is
the weight of the bandwidth cost. According to Eq. (2), if we increase \( \alpha \), the score of the video quality will increase while the score of the bandwidth cost will decline. Thus, as \( \alpha \) increases, the average score will be lifted. However, it is worth to note that it is unfair to compare average reward with different values of \( \alpha \). Increasing \( \alpha \) does not mean the improvement of algorithm performance. The objective should be maximizing average reward with any fixed \( \alpha \).

We verify this point through an experiment varying \( \alpha \) from 0.1 to 0.9. We set the buffer capacity with \( M = 70 \). Fig. 7 presents the average reward of our policy over different values of \( \alpha \) with \( AR = 0.11, 0.5, 0.9 \). As we see, the average reward increases as \( \alpha \) increases. The average reward increases from 0.045 to 6.426 for the case \( AR = 0.11 \), from 0.725 to 7.955 for the case \( AR = 0.5 \) and from 0.918 to 8.586 for the case \( AR = 0.9 \) when \( \alpha \) increases from 0.1 to 0.9. When \( \alpha \) is high, the term of QoE in reward function Eq. (12) is weighted heavier and thus VVUOP tends to make more efforts to improve user QoE. As \( \alpha \) declines, the cost becomes more importance and VVUOP will impress user QoE to save bandwidth cost. We can observe that by fixing \( \alpha \), our policy can achieve greater average reward when the AR is increased. More WiFi resources allow more video chunks to be uploaded via free WiFi networks, thus reducing the cost of cellular usage.
G. Impact of Location Information

Location information can be used to calculate the distance between a vehicle and RSUs to predict the WiFi bandwidth accurately. Therefore, we properly revise the original WiFi bandwidth model so that the WiFi bandwidth depends on the distance between the vehicle and its nearest RSU. We assume that the WiFi network is not available when the distance between the vehicle and its nearest RSU is greater than 70 meters, and the smaller the distance between the vehicle and its nearest RSU is, the higher the WiFi bandwidth is. To verify the performance of our revised model, we conduct a set of trace-driven simulations based on the routes of 40 buses over 20 days in Amherst Town center measured by [51]. The trace data contains GPS locations of buses, durations of connections with RSUs and so on. Fig. 8 shows a route of a tested bus and the deployment of RSUs. The buffer is set as $M = 40$ and Fig. 9 shows the average reward under three policies with objective tradeoff parameter $\alpha = 0.5$. The average reward of the optimal policy is calculated by dynamic programming with knowing all global information including the cellular and WiFi bandwidth of every slot and it serves as an upper bound of the performance. The policy Location + VVUOP is derived from the revised model which takes the location information into account. The last one is our original model without considering the location information. From the figure, we can find that Location + VVUOP achieves a higher average reward in comparison to VVUOP. But the improvement is limited, which shows the advanced nature of our model.

From above figures, we can summarize the features of different policies. MWUU is an
extremely conservative policy to minimize the usage of cellular networks. It can perform well when WiFi networks are resourceful or users significantly care about their monetary cost incurred by cellular traffic. Nevertheless, users may weight more on video quality and WiFi service could be very bad. In such situation, the performance of MWUU will be very bad. MMDU always leverages cellular networks to offload data. This policy will be punished severely if there is a heavy weight put on bandwidth cost. MRU tries to maximize the reward in each time slot. It can always make fairish decisions in different conditions. However, it fails to optimize the average reward from a long-term perspective with global information. Thus, MRU cannot surpass our policy VVUOP.

**VII. Conclusion**

In this paper, we study the cloud video content uploading via vehicle access networks (including WiFi and cellular networks) for moving vehicles. We propose to adaptively select the bitrate for chunks to be uploaded so as to maximize utility a metric defined as a tradeoff between bandwidth cost and video quality. By formulating the problem as a constrained MDP problem, we can analyze the condition to achieve the maximal utility. Based on the analysis, an iterative algorithm is devised. Simulation results validate the advantages of our policy compared with other candidates. In our model, we assume that WiFi service is free. Our proposed framework can be extended to the case when the WiFi network is not free, under which the objective
function (i.e., the average reward) can be modified slightly to take the WiFi cost into account. The framework created with MDP can still be applied to analyze this problem, and in this case it is not difficult to understand that users are prone to select lower bitrates for chunks so as to save bandwidth cost given other factors unchanged. In our future work, we intend to consider the scenario that a vehicle is moving according to the navigation route and explore the optimal upload policy of such scenario.

APPENDIX

A. Proof of Lemma 2

Proof: To prove Lemma 2, we introduce another criterion expected total discounted reward of decision function $f$ which can be defined as:

$$v_f^\beta = \sum_{t=0}^{\infty} \beta^t P_f^t R_f,$$  (30)

where $v_f^\beta$ is a vector representing the total discounted reward of each initial state. The above expression is also equivalent to

$$v_f^\beta = \sum_{t=0}^{\infty} \beta^t P_f^t R_f = R_f + \beta P_f v_f^\beta.$$  (31)

We define an operator $O_f^\beta(v) = R_f + \beta P_f v$. And we have the following lemma:

Lemma 3 If $v \leq O_f^\beta(v)$, then $v \leq v_f^\beta$.

Proof: There is:

$$v \leq R_f + \beta P_f v$$  (32)

$$\leq R_f + \beta P_f R_f + \beta^2 P_f^2 v.$$  

Generally, we have:

$$v \leq \sum_{t=0}^{T} \beta^t P_f^t R_f + \beta^{T+1} P_f^{T+1} v.$$  (33)

We let $T$ tend to infinity and we have:

$$v \leq v_f^\beta.$$  (34)
We define $H^\beta_f$ as:

$$H^\beta_f = \sum_{t=0}^{\infty} \beta^t (P^t_f - P^*_f).$$  \hspace{1cm} (35)

In fact, when $\beta$ tends up to 1, $H^\beta_f$ converges to a matrix. The following lemma will be leveraged by Lemma 5.

**Lemma 4** If $P$ is a real matrix and when $n \to \infty$, $P^n \to 0$, then inverse matrix of $I - P$ exists and:

$$(I - P)^{-1} = \sum_{n=0}^{\infty} P^n.$$  \hspace{1cm} (36)

**Proof:** For $\forall n > 1$, there is:

$$(I + P + P^2 + \cdots + P^{n-1})(I - P) = I - P^n.$$  \hspace{1cm} (37)

Due to $P^n \to 0$ as $n \to \infty$, when $n$ is large enough, $|I - P^n| \neq 0$. Thus $|I - P| \neq 0$ and therefore the inverse matrix of $I - P$ exists. We have:

$$\sum_{k=0}^{n-1} P^k = (I - P^n)(I - P)^{-1}.$$  \hspace{1cm} (38)

We let $n \to \infty$ and yield the result. \hspace{1cm} $\blacksquare$

**Lemma 5** When $\beta \uparrow 1$,

$$H^\beta_f \to H_f = (I - P_f + P^*_f)^{-1} - P^*_f.$$  \hspace{1cm} (39)

**Proof:** We first prove that for $\forall n > 0$, there is

$$P^n_f - P^*_f = (P^n_f - P^n_f)P_f.$$  \hspace{1cm} (40)

When $n = 1$, it is obvious. Suppose that Eq. (40) is satisfied for $n$, then

$$\begin{align*}
(P_f - P^*_f)^{n+1} &= (P_f - P^*_f)^n (P_f - P^*_f) \\
&= (P^n_f - P^*_f) (P_f - P^*_f) \\
&= P^n_f - P^*_f.
\end{align*}$$  \hspace{1cm} (41)
Thus expression Eq. (40) for \( \forall n > 0 \) is proven. There is:

\[
H_f^\beta = \sum_{n=0}^{\infty} \beta^n (P_f^n - P_f^*)
\]

\[
= \sum_{n=1}^{\infty} \beta^n (P_f - P_f^*)^n + I - P_f^*
\]

\[
= \sum_{n=0}^{\infty} \beta^n (P_f - P_f^*)^n - P_f^*. \tag{42}
\]

Because \( P_f \) and \( P_f^* \) are stochastic matrices and \( \beta \in (0, 1) \),

\[
\beta^n (P_f - P_f^*)^n \to 0, \quad n \to \infty. \tag{43}
\]

With Lemma 4, we know

\[
H_f^\beta = (I - \beta P_f + \beta P_f^*)^{-1} - P_f^*. \tag{44}
\]

We let \( \beta \) tend up to 1, and the result is yielded.

\[\blacksquare\]

**Lemma 6** For any decision function \( f \) and discounted factor \( \beta \in (0, 1) \), there is:

\[
v_f^\beta = \frac{g_f}{1 - \beta} + h_f + e(\beta), \tag{45}
\]

where \( e(\beta) \to 0 \) as \( \beta \uparrow 1 \).

**Proof:**

\[
v_f^\beta = \sum_{n=0}^{\infty} \beta^n P_f^n R_f
\]

\[
= \sum_{n=0}^{\infty} \beta^n (P_f^* + P_f^n - P_f^*) R_f
\]

\[
= \frac{P_f^* R_f}{1 - \beta} + H_f^\beta R_f
\]

\[
= \frac{P_f^* R_f}{1 - \beta} + h_f + (H_f^\beta - H_f) R_f
\]

\[
= \frac{P_f^* R_f}{1 - \beta} + h_f + e(\beta). \tag{46}
\]

\[\blacksquare\]
Now we turn back to our Lemma 2 and we have

\[
O_f'(v_f^\beta) - v_f^\beta = \left[ R_f' + \beta P_f' \left( \frac{g_f}{1-\beta} + h_f + e(\beta) \right) \right]
- \left( \frac{g_f}{1-\beta} + h_f + e(\beta) \right)
= \frac{1}{1-\beta} (P_f' g_f - g_f) - (P_f' g_f - R_f' + (I - \beta P_f') h_f)
+ \beta P_f' e(\beta) - e(\beta)
= \frac{1}{1-\beta} (P_f' g_f - g_f) - (P_f' g_f - R_f' + (I - P_f') h_f)
+ \beta P_f' e(\beta) - e(\beta) - (1 - \beta) P_f' h_f. \tag{47}
\]

If \( f' \) satisfies the Eq. (23), Eq. (47) tends up to infinity while \( \beta \) tends up to 1. Else if \( f' \) satisfies the Eq. (24) and Eq. (25), there is:

\[
O_f'(v_f^\beta) - v_f^\beta
= \frac{1}{1-\beta} (P_f' g_f - g_f) - (P_f' g_f - R_f' + (I - P_f') h_f)
+ \beta P_f' e(\beta) - e(\beta) - (1 - \beta) P_f' h_f
= -(g_f - R_f' + (I - P_f') h_f) + \beta P_f' e(\beta) - e(\beta)
- (1 - \beta) P_f' h_f. \tag{48}
\]

Eq. (48) is greater than or equal to 0 while \( \beta \) tends up to 1. Thus with Lemma 3, there is

\[
g_f' = \lim_{\beta \to 1^-} (1 - \beta)(v_f^\beta - h_f + e(\beta))
= \lim_{\beta \to 1^-} (1 - \beta)v_f^\beta
\geq \lim_{\beta \to 1^-} (1 - \beta)v_f^\beta
\geq g_f \quad \tag{49}
\]
B. Proof of Theorem 1

Proof: According to the precondition of Theorem 1, it implies that for \( \forall f \) decision function, there is:

\[
g_{f_*} \geq P_f g_{f_*} \quad (50)
\]

\[
g_{f_*} \geq R_f + (P_f - I) h_{f_*} \quad (51)
\]

Substituting Eq. (51) into Eq. (50) can yield:

\[
g_{f_*} \geq P_f g_{f_*} \geq P_f R_f + P_f (P_f - I) h_{f_*} \quad (52)
\]

We repeat the procedure by substituting the above expression into Eq. (50) and we can obtain:

\[
g_{f_*} \geq P_f g_{f_*} \geq P^2_f R_f + P_f (P_f - I) h_{f_*} \quad (53)
\]

Generally, we have:

\[
g_{f_*} \geq P^t_f R_f + P^t_f (P_f - I) h_{f_*} \quad (54)
\]

Summing these expression from \( t = 0 \) to \( t = T \) can yield:

\[
(T + 1)g_{f_*} \geq \sum_{t=0}^{T} P^t_f R_f + (P_f^{T+1} - I) h_{f_*} \\
\geq g_{f,T} + (P_f^{T+1} - I) h_{f_*} \quad (55)
\]

Since \( P_f^{T+1} \) is a stochastic matrix and \( h_f \) is limited, we can obtain

\[
\lim_{T \to \infty} \frac{(P_f^{T+1} - I) h_{f_*}}{T + 1} = 0. \quad (56)
\]

With above conclusion, we have

\[
g_{f_*} \geq \lim_{T \to \infty} \frac{g_{f,T}}{T + 1} \geq g_f \quad (57)
\]
C. Proof of Theorem 2

**Proof:** Recall that a general policy $\psi = (\pi_0, \pi_1, \cdots)$ consists of a series of decision rules (i.e. $\pi_t, t = 0, 1, \ldots$) and it employs the decision rule $\pi_t$ which choose an action with a probability distribution according to the state (i.e. $s(t)$) at time slot $t$.

The proof is similar to theorem 1. First, we prove that given the decision function $f_\ast$ obtained from Algorithm 1, for $\forall \psi = (\pi_0, \pi_1, \cdots) \in \Pi$, there is:

\begin{align}
g_{f_\ast} & \geq P_{\pi_t} g_{f_\ast} \quad (58) \\
g_{f_\ast} & \geq R_{\pi_t} + (P_{\pi_t} - I) h_{f_\ast} \quad (59)
\end{align}

where $t \in N^*$. Due to the optimal decision function $f_\ast$ satisfying theorem 1, it implies that for $\forall f$ decision function, there is:

\begin{align}
g_{f_\ast} & \geq P_f g_{f_\ast} \quad (60) \\
g_{f_\ast} & \geq R_f + (P_f - I) h_{f_\ast}. \quad (61)
\end{align}

We unfold the $i^{th}$ item of Eq. (60):

\[ g_{f_\ast}(s_i) \geq \sum_{j=1}^{|S|} P_f(s_j|s_i) g_{f_\ast}(s_j). \quad (62) \]

Due to the arbitrariness of $f$, the above expression can be represented as:

\[ g_{f_\ast}(s_i) \geq \sum_{j=1}^{|S|} P_{a_k}(s_j|s_i) g_{f_\ast}(s_j), \quad (63) \]

where $a_k$ is any available action on state $s_i$. We unfold the $i^{th}$ item of the right side of Eq. (58):

\[ \sum_{j=1}^{|S|} \sum_{k=1}^{|A|} P_{a_k}(s_j|s_i) \pi_t(a_k|s_i) g_{f_\ast}(s_j), \quad (64) \]

where $\pi_t(a_k|s_i)$ denotes the probability that decision rule $\pi_t$ chooses action $a_k$ given state $s_i$. Then we exchange the summation sign and we have:

\begin{align*}
\sum_{j=1}^{|S|} \sum_{k=1}^{|A|} P_{a_k}(s_j|s_i) g_{f_\ast}(s_j) \pi_t(a_k|s_i) \\
\leq \sum_{k=1}^{|A|} g_{f_\ast}(s_i) \pi_t(a_k|s_i) \\
\leq g_{f_\ast}(s_i). \quad (65)
\end{align*}
Thus Eq. (58) is followed. We can leverage the same trick to proof Eq. (59) and we leave it out here. Now applying Eq. (59) with \( t = 0 \), namely:

\[
g_f \geq R_0 + (P_0 - I)h_f.
\]  

(66)

Applying Eq. (58) with \( t = 0 \) and Eq. (59) with \( t = 1 \), we have:

\[
g_f \geq P_0 g_f \geq P_0 R_1 + P_0 (P_1 - I)h_f.
\]  

(67)

Similarly, we can obtain

\[
g_f \geq \prod_{\tau=0}^t P_{\tau} \geq P_0 \prod_{\tau=1}^t P_{\tau-1} R_{\tau} + \prod_{\tau=0}^t P_0 \prod_{\tau=1}^t P_{\tau-1} (P_\tau - I)h_f.
\]  

(68)

for \( \forall t \in \mathbb{N}^+ \). We denote \( \prod_{\tau=0}^t P_{\tau} \) by \( P_{\psi,t} \). By summing these inequation from \( t = 0 \) to \( t = T \), we can obtain:

\[
(T + 1)g_f \geq R_0 + \sum_{t=1}^T P_{\psi,t-1} R_{\tau} + (P_{\psi,T} - I)h_f
\]

\[
\geq g_{\psi,T} + (P_{\psi,T} - I)h_f,
\]  

(69)

where \( g_{\psi,T} \) represents the expected total reward vector over time slot \( 0 \) to time slot \( T \) by adopting policy \( \psi \). Since \( P_{\psi,T} \) is a stochastic matrix and \( h_f \) is limited, we can obtain

\[
\lim_{T \to \infty} \frac{(P_{\psi,T} - I)h_f}{T + 1} = 0.
\]  

(70)

With above conclusion, we have

\[
g_f \geq \limsup_{T \to \infty} \frac{g_{\psi,T}}{T + 1} \geq g_{\psi^*}.
\]  

(71)

\[\blacksquare\]

REFERENCES


Zhaobin Deng received the bachelors degree in information and computing science from Central South University, Changsha, China, in 2015. He is currently pursuing the master’s degree with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, under the supervision of Prof. D. Wu. His research interests mainly include multimedia networking and data center networking.

Yipeng Zhou is a lecturer in computer science with Department of Computing at Macquarie University, and the recipient of ARC Discovery Early Career Research Award, 2018. From Aug. 2016 to Feb. 2018, he was a research fellow at the Institute for Telecommunications Research (ITR) with University of South Australia. From 2013.9-2016.9, He was a lecturer with College of Computer Science and Software Engineering, Shenzhen University. He was a Postdoctoral Fellow with Institute of Network Coding (INC) of The Chinese University of Hong Kong (CUHK) from Aug. 2012 to Aug. 2013. He won his PhD degree supervised by Prof. Dah Ming Chiu and Mphil degree supervised by Prof. Dah Ming Chiu and Prof. John C.S. Lui from Information Engineering (IE) Department of CUHK in Aug. 2012 and Aug. 2008 respectively. He got Bachelor degree in Computer Science from University of Science and Technology of China (USTC) in July 2006.
Di Wu (M'06-SM'17) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2000, the M.S. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2007. He was a Postdoctoral Researcher with the Department of Computer Science and Engineering, Polytechnic Institute of New York University, Brooklyn, NY, USA, from 2007 to 2009, advised by Prof. K. W. Ross. He is currently a Professor and the Assistant Dean of the School of Data and Computer Science with Sun Yat-sen University, Guangzhou, China. His research interests include cloud computing, multimedia communication, Internet measurement, and network security. Prof. Wu was a co-recipient of the IEEE INFOCOM 2009 Best Paper Award. He has served as an Editor of the Journal of Telecommunication Systems (Springer), the Journal of Communications and Networks, Peer-to-Peer Networking and Applications (Springer), Security and Communication Networks (Wiley), and the KSII Transactions on Internet and Information Systems, and a Guest Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He has also served as the MSIG Chair of the Multimedia Communications Technical Committee in the IEEE Communications Society from 2014 to 2016. He served as the TPC Co-Chair of the IEEE Global Communications Conference—Cloud Computing Systems, and Networks, and Applications in 2014, the Chair of the CCF Young Computer Scientists and Engineers Forum—Guangzhou from 2014 to 2015, and a member of the Council of China Computer Federation.

Guoqiao Ye received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2016. He is currently pursuing the master degree in the Department of Computer Science, Sun Yat-sen University, Guangzhou, China, under the supervision of Prof. D. Wu. His research interests include smart grid, distributed computing and computer communications.

**Liang Xiao (M'09, SM'13)** received the B.S. degree in communication engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China; the M.S. degree in electrical engineering from Tsinghua University, Beijing, China; and the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 2000, 2003, and 2009, respectively. She is currently a Professor with the Department of Communication Engineering, Xiamen University, Fujian, China. Her current research interests include smart grids, network security, and wireless communications.