Efficient Upstream Bandwidth Multiplexing for Cloud Video Recording Services

Jian He, Di Wu, *Member, IEEE*, Xueyan Xie, Min Chen, *Senior Member, IEEE*, Yong Li, *Member, IEEE*, and Guoqing Zhang

***Abstract*— The upsurge of cloud video recording (CVR) has gained increasing attention from the general public and entrepreneurs. With live video records archived in the cloud, the CVR paradigm enables various smart services by keeping track of activities in the monitored region from anywhere at any time. However, the limited upstream bandwidth affects the quality of surveillance when multiple distributed cameras share the same upstream link. To solve the problem, this paper proposes an efficient upstream bandwidth multiplexing algorithm to intelligently allocate upstream bandwidth for each live video stream while maximizing the overall utility from the perspective of a CVR user. Specifically, we formulate the upstream bandwidth multiplexing problem as a constrained stochastic optimization problem, and apply the technique of hierarchical approximation to solve it efficiently. Our algorithm can be extended to take the priority of video streams into account and allocate more upstream bandwidth to video streams with higher priorities. We explicitly prove the approximation ratio of the proposed algorithm. In addition, we also conduct extensive trace-driven simulations to verify the effectiveness of our algorithm. The simulation results show that our algorithm improves the overall CVR user utility by over 20% compared with other alternatives, and the average utility per bandwidth unit is guaranteed to be stable even when the number of video streams increases.**

***Index Terms*— Adaptive streaming, allocation optimization, bandwidth multiplexing, cloud video recording (CVR).**

I. Introduction

TRADITIONALLY, closed-circuit television equipments or digital video recorders (DVRs) were widely employed to perform video surveillance over a region [1]. Video streams are recorded directly to network-attached storage or internal flash storage. However, the cost of purchasing specialized cameras and recording devices is relatively expensive and such solution is also not scalable when a huge amount of video streams should be stored, processed and delivered [2]. With the advance of cloud computing technology, cloud video recording (CVR) is emerging as a promising solution to deliver affordable video surveillance services for common users.

Different with conventional video surveillance approaches, CVR-based solution first needs to transmit live video streams generated by wireless cameras (e.g., webcam and IP camera) to the cloud, so that the footage will never be lost and always be accessible. On the other hand, the cloud is able to securely stream the video to authorized users, and perform intelligent analysis over the footage and send alerts to a CVR user when a specific event has been detected in a monitored region. Various CVR users can also create video clips from their footage and share with their friends and family easily.

The potential of CVR has also attracted significant attention from industrial practitioners, ventures, and researchers. A few leading competitors in the market include DropCam [3], iVideon [4], and Cloud DVR [5]. On June 20, 2014, DropCam was acquired by Google's Nest Labs with U.S. $555 million. The market of CVR is expected to keep expanding in the next few years [6]. However, the practical development of CVR systems is still in its infancy and there are a number of technical challenges to be addressed.

Given a typical scenario in an indoor environment (e.g., building, house, and apartment), a CVR user normally deploys multiple wireless cameras in the monitored region. All video streams generated by those wireless cameras are transmitted simultaneously via wireless access points (APs) to the remote cloud server. There exist many unique challenges to design such a new service architecture. As each video stream has different utility to the CVR user, it is important to properly allocate upstream bandwidth among video streams to achieve utility maximization. Meanwhile, because it is not easy to predict which video stream will later be viewed by the CVR user, we also need to ensure that the received utility of each video stream is above a minimum threshold, so as to avoid missing all details in any video stream. Different from the utility analysis for a single video stream, the utility of a CVR user is determined by the total utility of multiple video streams. A straightforward approach is to share the upstream bandwidth among multiple cameras in a time-division manner [7]. However, such a simple approach is hard

Manuscript received May 20, 2015; revised July 8, 2015; accepted August 22, 2015. Date of publication September 2, 2015; date of current version September 30, 2016. This work was supported in part by the National Science Foundation of China under Grant 61174152, Grant 61272397, Grant 61331008, and Grant 61572538, in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant S20120011187, in part by the Basic Research Program of China (973 Program) under Grant 2011CB302505, and in part by the Innovation Team through Guangdong Scientific Bureau, China, under Grant 201001D0104726115. This paper was recommended by Associate Editor J. Liang. *(Corresponding author: Di Wu.)*

J. He, D. Wu, and X. Xie are with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China, and also with the SYSU–CMU Shunde International Joint Research Institute, Foshan 528300, China (e-mail: isshejian@gmail.com; wudi27@mail.sysu.edu.cn; xiexy9@mail2.sysu.edu.cn).

M. Chen is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: minchen@ieee.org).

Y. Li is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: liyong07@tsinghua.edu.cn).

G. Zhang is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: gqzhang@ict.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2475875

to ensure the optimality and stability of user utility when the bandwidth resource demand of live video streams fluctuates with time. The objective problem, normally considered as a constrained stochastic utility maximization problem, has very high computational complexity incurred by video-layer selection, flow heterogeneity, and temporal utility fluctuations. It is very challenging to design an efficient algorithm without using any prediction techniques.

In this paper, we aim at optimizing the overall CVR user utility by efficient upstream bandwidth multiplexing among multiple cameras for CVR services. To this end, we formulate the problem as a constrained stochastic joint optimization problem of bandwidth allocation and rate adaptation. This optimization problem can be further decomposed into two subproblems, namely, bandwidth allocation problem and rate adaptation problem. The former determines how much upstream bandwidth should be provisioned for each video stream, while the later decides which video layers should be transmitted. However, the decomposed problems are proven to be NP-hard. Thus, we further propose an integer-approximation algorithm to solve the decomposed problems and explicitly prove the approximation ratio of our proposed hierarchical approximation algorithm. Overall, our main contributions of this paper can be summarized as below.

1) To the best of our knowledge, we are the first to study the problem of upstream bandwidth multiplexing for cloud-based video recording services. We formulate the CVR user utility maximization problem as a constrained stochastic optimization problem, and consider both flow multiplexing and temporal multiplexing. Our formulation is able to be extended to take different priorities of video streams into account.
2) We decompose the optimization problem into two subproblems and explicitly prove their NP-hardness. Then, we propose a hierarchical approximation algorithm to maximize the CVR user utility. Our algorithm has low computational complexity and also avoids the use of any prediction techniques. Moreover, it does not require the feedback on received video frames from the viewing side.
3) We conduct extensive trace-driven simulations to evaluate the effectiveness of our proposed algorithm. The simulation results show that our algorithm can improve the overall CVR user utility by over 20% compared with other approaches, and ensure that the average utility per bandwidth unit remains at a high level even when the number of video streams increases. We also examine the adaptability of our proposed algorithm under different user preference settings.

The rest of this paper is organized as follows. Section II reviews related work in the area of CVR and surveillance. Section III describes the system model and problem formulation. Section IV describes the proposed algorithm for upstream bandwidth multiplexing. Section V provides the simulation results for evaluating the effectiveness of our algorithm. Finally, Section VI concludes this paper and discusses future directions.

II. Related Work

The popularity of CVR is attributed to the technical development in multiple domains, including video sensor network (VSN), wireless video transmission, Internet video streaming, and cloud computing.

VSNs have been extensively studied in [8]. Previous works mostly focused on resource provisioning to enable efficient monitoring in the targeted area, scheduling and placement of video sensor nodes, workload balancing on sensor nodes, and reliable communication support for distributed sensor nodes. To address these problems, researchers investigated various solutions for video tracking [9], video surveillance [10], network architecture [11], communication protocols [12], and video compression [13]. In spite that a CVR system is similar to a VSN, the CVR system differs in that it has a powerful back-end cloud to deliver intelligent services which is much more scalable.

A CVR system normally relies on wireless links to transmit video streams from cameras to nearby APs. Previous works on wireless video transmission focused on the challenges incurred by stochastic wireless network conditions, for example, reliable video transmission over wireless channels [14], real-time wireless video transmission [15], and energy-efficient wireless communications [16]. To the best of our knowledge, none of them investigated the bandwidth resource multiplexing problem across video streams sharing the same uplink.

Another approach to address network condition variations is to control the playback rate of a video stream adaptively. Related techniques include scalable video coding (SVC) [17] and dynamic adaptive streaming over HTTP (DASH) [18], [19]. If a video stream is coded with SVC, then the rate can be controlled by selecting different video layers to be transmitted [20]. A DASH-based system can encode one video chunk into multiple versions with different playback rates and store them as separate files. Thus, a DASH client can switch the rate by requesting different video files [18]. Different metrics (e.g., network condition and playback buffer state) can be adopted in designing the rate adaptation algorithms. Prediction techniques were often used to increase the effectiveness of rate adaptation (see [20]–[22]).

Internet video streaming is also a closely related area. To guarantee that each video stream has sufficient resource, Ren and van der Schaar [23] proposed to allocate resources to each video stream in a time-division manner. Ma *et al.* [24] considered to utilize adaptive streaming to improve streaming performance. The rate of a video stream can be dynamically controlled based on the network condition [25], user quality of experience (QoE) [19], and playback buffer status [26]. The huge resource demand of delivering video streams has increasingly exerted pressures on the underlying video distribution system. The researchers have proposed various system architectures to reduce the deployment cost while ensuring user QoE. One promising approach is to introduce cloud computing technologies and leverage the power of cloud platforms to enable large-scale video streaming (see [27], [28]). Wen *et al.* [29] summarized the emerging paradigms of cloud mobile media and proposed a cloud-centric media platform for cloud-based mobile media. Jin *et al.* [30] further leveraged

TABLE I
NOTATIONS

Notation	Description
N	the number of cameras
$\mathcal{C}$	the set of cameras
τ	the length of a time slot
$f_t^{c,b}$	the base-layer video frame generated by camera c in time slot t
$m_t^{c,b}$	the number of packets of the frame $f_t^{c,b}$
$\mathcal{E}$	the set of enhancement layers
K	the number of enhancement layers
$f_t^{c,e}$	the set of packets corresponding to the enhancement-layer e video frame generated by camera c in time slot t
$m_t^{c,e}$	the number of packets of the frame $f_t^{c,e}$
B	the maximal upstream capacity of AP
$u(f_t^{c,b})$(or $u(f_t^{c,e})$)	the utility that the frame $f_t^{c,b}$(or $f_t^{c,e}$) carries
$D_c^b(t)$	the decision for base-layer queue of camera c in time slot t
$D_c^e(t)$	the decision for enhancement-layer e queue of camera c in time slot t
$b_c(t)$	transmission rate allocated to camera c in time slot t
u_0	the minimum utility threshold
l	the size of a video packet
$\bar{u}_c(T)$	the average utility of camera c over T time slots
$H_c(t)$	the size of virtual queue of camera c in time slot t
V	the tuning parameter in Lyapunov optimization framework

the media cloud to deliver on-demand adaptive video streaming services, and achieved a three-way tradeoff between the caching, transcoding, and bandwidth cost, at each edge server. There are some other works to study the design of video transmission schemes over different kinds of architectures (see [31]–[33]).

Compared with previous works, our work is the first to address the problem of upstream bandwidth multiplexing in CVR systems. Our problem formulation takes the overall utility of multiple video streams into account, which is different from that in [7] and [34]–[36]. In the design of a CVR system, the main challenge lies in that the received user utility cannot be known beforehand, namely, it is hard to learn which video stream will later be viewed by a CVR user. Unlike traditional approaches, our solution does not require any predicted information and thus avoids the complexity incurred by inaccurate prediction.

III. System Model and Problem Formulation

In this section, we will describe our system model and formulate the investigated problem in detail. All notations used in this paper are summarized in Table I.

A. System Model

Let us consider a typical CVR system, as shown in Fig. 1, in which a CVR user (i.e., an organization or individual) can deploy a set of wireless cameras to perform real-time surveillance over a region (e.g., building, townhouse, and apartment). These wireless cameras can be placed at well-chosen locations to increase the coverage. All video streams

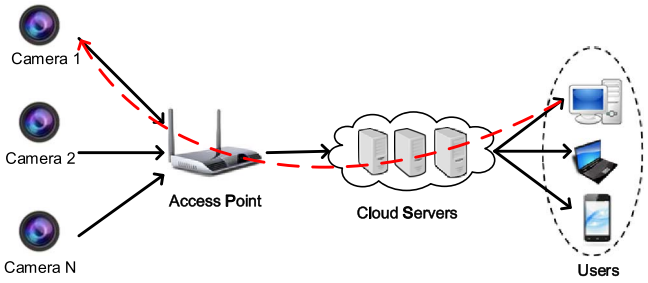


Fig. 1. Functional components in a typical CVR system: cameras, AP, cloud servers, and users.

captured by the cameras are transmitted over wireless links to the nearby AP, which further uploads video streams to the remote video cloud service provider (e.g., DropCam). As the upstream link of the AP has limited capacity, the upstream link has to be multiplexed by different video streams. After receiving raw video streams, the process chain is performed in video cloud in terms of content storing, video transcoding, and video content delivery. With the assistance of the cloud, any authorized user can watch a live stream of any camera or replay recorded videos at any time on any devices.

Although the single-AP model considered in this paper can cover most use cases in current CVR systems, it is possible that multiple APs may be accessible in a specific region. In this case, we can divide a large region into multiple small regions according to the coverage of each AP. If multiple APs share the same upstream link, we can treat those APs as a single virtual AP and still be able to characterize key features using the single-AP system model. However, if multiple APs belong to different CVR users and share the upstream bandwidth in an uncooperative manner, it is necessary to use other system models since we should solve a completely different objective problem for bandwidth sharing. The problem of bandwidth allocation among uncooperative users is beyond the scope of this paper. We will investigate the multi-AP case of the bandwidth multiplexing problem in the future work.

There are two kinds of data flows in our system, namely, *video data flow* and *control data flow*. The video data flows are transferred to cloud servers via AP, and further delivered to viewers after processing, while the control flows are transmitted in the reverse direction. A CVR user can control the transmission behaviors of cameras by sending performance requirements (e.g., basic utility constraint) for video streams. The bandwidth allocation algorithm is running on the AP such that cameras will transmit video streams according to resource allocation decisions made by the AP.

A CVR user needs to pay the cloud service cost (including storage and transcoding service) to a CVR service provider, and the bandwidth cost to the Internet service provider. How to better utilize the limited upstream bandwidth to achieve high-utility video recording is the focus of our research.

Suppose that the system operation is time slotted, and each time slot lasts for τ. The set of cameras is denoted by $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$. Also assume that cameras utilize SVC video encoding techniques to process their video bitstreams before transmitting them to the AP. Each video

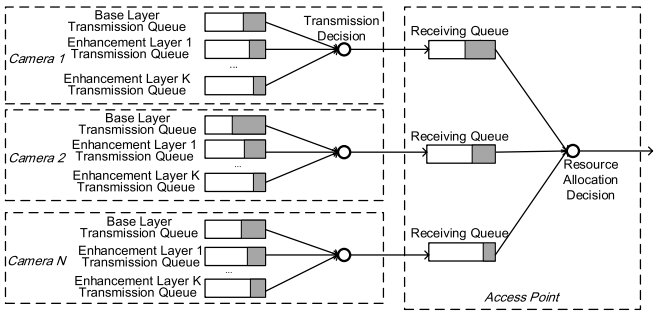


Fig. 2. Multilayer video transmissions between distributed cameras and the AP.

frame consists of multiple video layers, including one base layer and multiple enhancement layers. In each time slot t, each camera c generates a video frame $f_t^{c,b}$ corresponding to the base layer, which consists of a set of video packets $f_t^{c,b} \doteq \{p_t^{c,b}(1), p_t^{c,b}(2), \ldots, p_t^{c,b}(m_t^{c,b})\}$, where $m_t^{c,b}$ represents the number of video packets associated with video frame $f_t^{c,b}$. Denote the set of enhancement layers by $\mathcal{E}$ with $|\mathcal{E}| = K$. For an enhancement layer $e \in \mathcal{E}$, each camera c also generates a video frame $f_t^{c,e}$, denoted by $f_t^{c,e} \doteq \{p_t^{c,e}(1), p_t^{c,e}(2), \ldots, p_t^{c,e}(m_t^{c,e})\}$. The dependency among enhancement layers can be represented by the relation $\prec$. Specifically, if $e_1 \prec e_2, e_1, e_2 \in \mathcal{E}$, then the enhancement layer e_2 cannot be decoded unless the enhancement layer e_1 has been received.

The transmissions between distributed cameras and the AP are described in Fig. 2. Each camera is configured with a base-layer transmission queue and multiple enhancement-layer transmission queues, which buffer the packets corresponding to base-layer frames and enhancement-layer frames, respectively. As a CVR user usually views live video streams, the transmission queues only need to store recently generated video packets. To ensure the timeliness of video transmission, the size of queues on either cameras or APs is set to a video frame, as a large queue will introduce significant delay. Note that the number of packets associated with each video frame $m_t^{c,b}$ (or $m_t^{c,e}$) depends on the amount of information contained in the frame. In the AP, each receiving queue buffers video packets generated by a camera. The AP cannot know beforehand whether a packet belongs to a base layer or an enhancement layer without performing decoding. For simplicity, we assume that the AP allocates one receiving queue for each camera.

The decision on resource allocation made by an AP adjusts data rate of each camera, while the transmission decision indicates which queues packets will enter. After receiving resource allocation decisions, each camera makes transmission decisions in the following way: selecting packets from lower layers until the bandwidth resource needed to transmit these selected packets reaches the amount determined by resource allocation decisions.

To ensure liveness of video streams, each video packet is associated with a transmission deadline. All video packets belonging to the same video frame have the same deadline. Note that the transmission deadline depends on the frame rate of video streams generated by cameras, denoted by $1/\tau$, which indicates that each camera will generate one video frame during each time slot. Moreover, we assume that packets of one video frame cannot provide any utility for a user unless all packets of that video frame can be successfully transmitted to the user. The utility that a video frame f_t^c carries is defined by a function $u(f_t^c)$. Without loss of generality, we assume that $u(f_t^{c,e}) \leq u(f_t^{c,b}), \forall e \in \mathcal{E}$, which captures the dependency between the base-layer video frame and enhancement-layer video frames. Intuitively, the utility of a video stream is a measure of user QoE on video quality assessment.

The total transmission rate of an AP is constrained by B. Denote the transmission rate allocated to receiving queue c by $b_c(t)$, then the amount of video which can be transmitted in one time slot for receiving queue c is $b_c(t) \cdot \tau$. The transmission decision of camera c is defined by a vector $(D_c^b(t), D_c^{e_1}(t), D_c^{e_2}(t), \ldots, D_c^{e_K}(t))$, in which $D_c^b(t), D_c^{e_k}(t) \in \{0, 1\}, e_k \in \mathcal{E}$. If $D_c^b(t)$ equals to 1, then all packets in the base-layer transmission queue of camera c will be transmitted to the AP in time slot t; otherwise no packets in this queue will be transmitted. And $D_c^{e_k}(t)$ corresponds to the transmission decision to the enhancement-layer transmission queue of camera c.

B. Problem Formulation

Denote the time-average user utility obtained by transmitting the video stream from a camera c by $\bar{u}_c(T) = (1/T)\sum_{t=1}^{T}[u(f_t^{c,b})D_c^b(t) + \sum_{e_k \in \mathcal{E}} u(f_t^{c,e})D_c^{e_k}(t)]$. Our objective is to maximize the overall time-average utility from the perspective of a CVR user. Thus, we can formulate it into the following optimization problem:

$$
\begin{aligned}
\mathbf{P1}: \quad & \max \lim_{T\to\infty} \sum_{c\in\mathcal{C}} \bar{u}_c(T) \\
& \text{s.t.} \sum_{c\in\mathcal{C}} b_c(t) \leq B \quad \forall t \quad (a) \\
& D_c^{e_k}(t) \leq D_c^b(t) \quad \forall c \in \mathcal{C}, e_k \in \mathcal{E}, t \quad (b) \\
& D_c^{e_2}(t) \leq D_c^{e_1}(t) \quad \forall c \in \mathcal{C}, e_1 \prec e_2, t \quad (c) \\
& \left(m_t^{c,b} D_c^b(t) + \sum_{e_k\in\mathcal{E}} m_t^{c,e_k} D_c^{e_k}(t)\right) \cdot l \leq b_c(t)\tau \quad (d) \\
& \lim_{T\to\infty} \bar{u}_c(T) \geq u_0 \quad \forall c \in \mathcal{C} \quad (e) \\
& D_c^{e_k}(t) \in \{0, 1\} \quad \forall c \in \mathcal{C}, e_k \in \mathcal{E}, t \quad (f) \\
& D_c^b(t) \in \{0, 1\} \quad \forall c \in \mathcal{C}, t \quad (g) \\
& 0 \leq b_c(t) \leq B \quad \forall c \in \mathcal{C}, t \quad (h) \\
\text{variables: } & b_c(t), D_c^b(t), D_c^{e_k}(t) \quad \forall c \in \mathcal{C}, e_k \in \mathcal{E}, t.
\end{aligned}
$$

In the above formulated problem, constraint (a) indicates the upper bound for the AP transmission rate. Constraint (b) captures the dependency between the base-layer video frame and the enhancement-layer video frames, while constraint (c) shows the dependency relationship $\prec_{\mathcal{E}}$ among enhancement layers. Constraint (d) indicates that all chosen packets from a camera c can be transmitted by the AP in time slot t, where l denotes the size of video packet. Constraint (e) is used to ensure that the minimum utility of each camera is over a threshold u_0 for fairness among all cameras.

Note that wireless cameras are connected with the AP via wireless channels, while the AP itself is normally connected with the Internet through wireline networks (e.g., asymmetric digital subscriber line and Ethernet). Thus, the maximal upstream bandwidth capacity of the AP is set to a fixed value B. In general, wireless cameras can have abundant transmission bandwidth to the AP due to highly developed WLAN techniques (such as 802.11n). Even if the available transmission bandwidth of a wireless camera varies significantly due to channel condition fluctuations, we can instead aim to maximize the time-average total expected utility. In this case, the utility of a camera c in time slot t can be defined as the weighted sum of $u_c(t)$, where the weights are directly obtained from the probability distribution of channel conditions. Thus, we remove wireless channel conditions from the problem formulation, but the algorithm design still follows the same scheme since the expected utility case does not change the properties of the objective function.

IV. Design of Online Resource Allocation Algorithm for Video Stream Multiplexing

A. Problem Decomposition

To avoid the difficulty of user demand prediction, we apply the Lyapunov optimization framework to design online algorithms and solve the optimization problem **P1**. Lyapunov optimization technique can satisfy the time-average constraints by maintaining the status of a set of virtual queues. The stochastic optimization problem can be converted to a set of one-shot optimization problems, which maximize a weighted sum of the instantaneous objective function and the size of virtual queues. Even without predicting the value of the objective function, Lyapunov optimization-based algorithm can provide a bounded approximation to the optimal solution.

We first introduce a set of virtual queues $\mathbf{H} = \{H_c, c \in \mathcal{C}\}$ to mitigate the computational complexity brought by the time-average constraint (e) in **P1**. The update on queue occupancy is defined as follows:

$$H_c(t+1) = \max[H_c(t) - u_c(t) + u_0, 0]$$

where $u_c(t) = u(f_t^{c,b})D_c^b(t) + \sum_{e\in\mathcal{E}} u(f_t^{c,e})D_c^e(t)$ denotes the CVR user utility of a camera c in time slot t. Then, we can derive the following lemma.

Lemma 1: Constraint (c) in the optimization problem **P1** can be equivalently transformed as below

$$\lim_{t\to\infty} \frac{H_c(t)}{t} \le 0 \quad \forall c \in \mathcal{C}.$$

Proof: From the queue update, we can observe that

$$\begin{aligned} H_c(t+1) &= \max[H_c(t) - u_c(t) + u_0, 0] \\ &\ge H_c(t) - u_c(t) + u_0. \end{aligned}$$

Summing the inequality over T time slots, then we have

$$H_c(t) - H_c(1) \ge T \cdot u_0 - \sum_{t=1}^{T} u_c(t).$$

Dividing both sides of the above inequality by T, we have

$$\frac{H_c(T)}{T} - \frac{H_c(1)}{T} \ge u_0 - \frac{1}{T}\sum_{t=1}^{T} u_c(t).$$

Calculating the time limit of the above inequality, we have

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} u_c(t) \ge u_0 - \lim_{T\to\infty} \frac{H_c(T)}{T} + \lim_{T\to\infty} \frac{H_c(1)}{T}.$$

Since $H_c(1)$ is a constant, the constraint $\lim_{T\to\infty} \bar{u}_c \ge u_0$ is equivalent to $\lim_{t\to\infty}(H_c(t)/t) \le 0$, which proves the lemma. ∎

Define the Lyapunov function as $L(t) = (1/2)\sum_{c\in\mathcal{C}} H_c^2(t)$. Then, the one-shot evolution of the Lyapunov function $\Delta(L(t))$ satisfies the following condition:

$$\begin{aligned} \Delta(L(t)) &= E[L(t+1) - L(t)|\mathbf{H}(t)] \\ &\le \sum_{c\in\mathcal{C}} \frac{1}{2}\left[(H_c(t) - u_c(t) + u_0)^2 - H_c^2(t)|\mathbf{H}(t)\right] \\ &\le \sum_{c\in\mathcal{C}} \left[(u_0 - u_c(t))H_c(t) + \frac{1}{2}(u_0 - u_c(t))^2|\mathbf{H}(t)\right] \\ &\le \sum_{c\in\mathcal{C}} [(u_0 - u_c(t))H_c(t) + (u_0 - u_{\max})^2|\mathbf{H}(t)] \\ &= \sum_{c\in\mathcal{C}} [(u_0 - u_c(t))H_c(t)|\mathbf{H}(t)] + N\cdot(u_0 - u_{\max})^2 \end{aligned}$$

where $u_{\max}$ denotes the maximum utility of all cameras. By exploiting the Lyapunov optimization framework, the optimization problem **P1** can be transformed into the following one-shot problem:

$$\begin{aligned} \mathbf{P2}\text{: } &\min \sum_{c\in\mathcal{C}} (u_0 - u_c(t))H_c(t) - V\cdot u_c(t) \\ &\text{s.t. } (a)(b)(c)(d)(f)(g)(h). \end{aligned}$$

Denote the decision that a camera c chooses at time slot t by $D_c(t) = (D_c^b(t), D_c^{e_1}(t), D_c^{e_2}(t), \ldots, D_c^{e_K}(t))$, $e_k \in \mathcal{E}, k \in [1, K]$, and the set of all decisions as $\mathbf{D}_c(t) = \{D_c(t)|(b)(c)(f)(g)\}$. Each camera c can choose only one decision $D_c(t)$ from the set $\mathbf{D}_c(t)$ at each time slot. Define $x(D_c(t)) \in \{0, 1\}$ indicating whether camera c chooses the decision $D_c(t)$. Note that transmission decisions of a camera are obtained from resource allocation decisions made by the AP. Then, we can transform the constraints (b), (c), (f), and (g) in the optimization problem **P2** into the following simple constraint:

$$\sum_{D_c(t)\in\mathbf{D}_c(t)} x(D_c(t)) \le 1 \quad \forall c \in \mathcal{C}. \quad (i)$$

Lemma 2: The one-shot optimization problem **P2** is NP-hard.

Proof: To prove the NP-hard property of the problem **P2**, we aim to build a polynomial-time reduction to **P2** from the knapsack problem which has been proven to be NP-hard.

The general knapsack problem is defined as follows:

$$\max_{\mathbf{x}} \sum_{i=1}^{n} v_i x_i$$
$$\text{s.t.} \sum_{i=1}^{n} \omega_i x_i \le W$$
$$x_i \in \{0,1\}.$$

Denoting $u_c(t) = u_c(D_c(t))$, we can map the variables in **P2** to parameters in the above problem as follows:

$$v_i = -u_0 H_c(t) + (V + H_c(t)) u_c(D_c(t))$$
$$x_i = x(D_c(t)) \quad \forall c \in \mathcal{C}, D_c(t) \in \mathbf{D}_c(t)$$
$$\omega_i = \left(m_t^{c,b} D_c^b(t) + \sum_{e \in \mathcal{E}} m_t^{c,e} D_c^e(t) \right) \cdot l$$
$$W = B.$$

Thus, it is clear that there exists a polynomial-time reduction algorithm. Problem **P2** has an additional constraint (i). Since the knapsack problem is NP-hard, then Problem **P2** must also be NP-hard. ■

To reduce the computational complexity of the one-shot optimization problem **P2**, we use an approximate approach to obtain good-enough solutions. Denoting $\omega_c(D_c(t)) = (m_t^{c,b} D_c^b(t) + \sum_{e \in \mathcal{E}} m_t^{c,e} D_c^e(t)) \cdot l$, and $\tilde{u}_c(D_c(t)) = -(u_0 H_c(t) - (V + H_c(t)) u_c(D_c(t)))$, we simplify the representation of Problem **P2** as follows:

$$\textbf{P3}: \max \sum_{c \in \mathcal{C}} \sum_{D_c(t) \in \mathbf{D}_c(t)} \tilde{u}_c(D_c(t)) x(D_c(t))$$
$$\text{s.t.} \sum_{D_c(t) \in \mathbf{D}_c(t)} x(D_c(t)) \le 1 \quad \forall c \in \mathcal{C}$$
$$\sum_{c \in \mathcal{C}} \sum_{D_c(t) \in \mathbf{D}_c(t)} \omega_c(D_c(t)) x(D_c(t)) \le B\tau$$
$$x(D_c(t)) \in \{0,1\} \quad \forall c \in \mathcal{C}.$$

To overcome the NP-hardness of Problem **P3**, we first relax the last constraint to the following one:

$$x(D_c(t)) \ge 0 \quad \forall c \in \mathcal{C}.$$

Then, we introduce two sets of dual variables $\lambda = \{\lambda_c, c \in \mathcal{C}\}$ and μ for the first two constraints, and we can obtain a dual problem of **P3** as follows:

$$\textbf{P4}: \min \sum_{c \in \mathcal{C}} \lambda_c + B\tau \cdot \mu$$
$$\text{s.t. } \lambda_c + \omega_c(D_c(t))\mu \ge \tilde{u}_c(D_c(t))$$
$$\forall D_c(t) \in \mathbf{D}_c(t), \quad c \in \mathcal{C}$$
$$\mu \ge 0, \quad \lambda_c \ge 0 \quad \forall c \in \mathcal{C}.$$

The problem transformation process can be summarized as follows. First, the original optimization problem **P1** is transformed to the one-shot optimization problem **P2** based on the Lyapunov optimization framework. Next, Problem **P3** is formulated as a relaxed version of **P2** so that it is simpler to find approximate algorithms to solve the problem. To design an approximate algorithm for the NP-hard problem **P3**, **P4** is the dual problem of **P3**.

Algorithm 1 Approximate One-Shot Algorithm

Input:
Bandwidth capacity B;
Decision set $\mathbf{D}_c(t), \forall c \in \mathcal{C}$;
Queue state $\mathbf{H}(t)$;
Tradeoff parameter V;
Utility threshold u_0;
Utility function $u_c(D_c(t)), \forall c \in \mathcal{C}, D_c(t) \in \mathbf{D}_c(t)$;
Bandwidth requirement function:
$\omega_c(D_c(t)), \forall c \in \mathcal{C}, D_c(t) \in \mathbf{D}_c(t)$.

Output:
Transmission decision $D_c(t), \forall c \in \mathcal{C}$.

1: Initialization step: $x(D_c(t)) = 0, \forall c \in \mathcal{C}, D_c(t) \in \mathbf{D}_c(t), \lambda_c = 0, \forall c \in \mathcal{C}$, $\mu = 1$, $u(t) = 0$, $\mathcal{Q} = \emptyset$, $\Delta = \min_{c \in \mathcal{C}, D_c(t) \in \mathbf{D}_c(t)} \frac{B\tau}{\omega_c(D_c(t))}$, $\Gamma = \max_{c \in \mathcal{C}, D_c(t) \in \mathcal{D}_c(t)} \omega_c(D_c(t))$.
2: **while** $\mu < \exp(\Delta - 1)$ AND $\mathcal{Q} \neq \mathcal{C}$ **do**
3: **for** $\forall c \in \mathcal{C} \setminus \mathcal{Q}$ **do**
4: $D_c^*(t) = \arg\max_{D_c(t) \in \mathbf{D}_c(t)} \tilde{u}_c(D_c(t))$;
5: **end for**
6: $c^* = \arg\max_{c \in \mathcal{C} \setminus \mathcal{Q}} \{\frac{\tilde{u}_c(D_c^*(t))}{\omega_c(D_c^*(t))\mu}\}$;
7: $x_{c^*}(D_{c^*}^*(t)) = 1$, $\lambda_{c^*} = \tilde{u}_{c^*}(D_{c^*}^*(t))$;
8: $u(t) = u(t) + \tilde{u}_{c^*}(D_{c^*}^*(t)), \mathcal{Q} = \mathcal{Q} \bigcup \{c^*\}$;
9: $\mu = \mu \cdot (\exp(\Delta - 1))^{\omega_{c^*}(D_{c^*}^*(t))/(B\tau - \Gamma)}$;
10: **end while**

To reduce the computational complexity of solving the original problem, we design an approximate algorithm to obtain solutions to Problem **P4**. The details of our proposed algorithm are shown in Algorithm 1.

Lemma 3: The solution achieved by Algorithm 1 is feasible to Problem **P3**.

Proof: We first examine whether the solution satisfies the first constraint and the third constraint. For the first constraint, line (4) and line (8) indicate that each selected camera c^* will be chosen once and only one corresponding decision $D_c^*(t)$ will be made. Thus, it is obvious that $\sum_{D_{c^*}(t) \in \mathbf{D}_{c^*}(t)} x(D_{c^*}(t)) = 1$. For the cameras c that are not chosen by Algorithm 1, it is true that $\sum_{D_c(t) \in \mathbf{D}_c(t)} x(D_c(t)) = 0$. For the third constraint, all the $x(D_c(t))$ are initialized to 0, and line (7) indicates that the value of $x(D_c(t))$ can only be updated to 1.

Then, for the second constraint, if the set of selected cameras in the previous $(r-1)$ rounds is denoted by $\tilde{\mathcal{C}}$, and the chosen camera in the rth round is c'. If the following condition is true, we have:

$$\sum_{c \in \tilde{\mathcal{C}}} \omega_c\left(D_c^*(t)\right) \le B\tau$$
$$\omega_{c'}\left(D_{c'}^*(t)\right) + \sum_{c \in \tilde{\mathcal{C}}} \omega_c\left(D_c^*(t)\right) \ge B\tau.$$

Then, since the maximum bandwidth requirement of a single camera cannot exceed the bandwidth capacity, that is, $\omega_{c'} \le \Gamma < B\tau$, we can derive

$$\sum_{c \in \tilde{\mathcal{C}}} \omega_c\left(D_c^*(t)\right) \ge B\tau - \Gamma.$$

Thus, it is true that

$$\frac{\sum_{c\in\tilde{\mathcal{C}}}\omega_c\left(D_c^*(t)\right)}{B\tau-\Gamma}\geq 1.$$

The value of μ^{r-1} in the $r-1$th round is calculated as follows:

$$\begin{aligned}\mu_{r-1}&=(\exp(\Delta-1))^{\frac{\sum_{c\in\tilde{\mathcal{C}}}\omega_c(D_c^*(t))}{B\tau-\Gamma}}\\&\geq\exp(\Delta-1)\end{aligned}$$

which indicates that the iteration will stop at the end of the $(r-1)$th round, and the camera c' will not be selected. Therefore, the second constraint is satisfied. ■

Note that the definition of utility function $u_c(D_c(t))$ depends on multiple factors, including the information contained in a video frame and the transmission cost of receiving a video frame. The maximal size of decision set $\mathbf{D}_c(t)$ is $|\mathcal{E}|+1$. Even using a brutal search method, the computational complexity of line (4) is $O((|\mathcal{E}|+1)\cdot N)$. Moreover, the number of iterations is upper bounded by the number of cameras N, thus the overall computation complexity of Algorithm 1 is $O((|\mathcal{E}|+1)\cdot N^2)$, which indicates that Algorithm 1 is a polynomial-time algorithm.

Next, we will continue to derive the approximation ratio of Algorithm 1. Although Lemma 3 ensures the feasibility of the solution to the primal variables of the problem **P3**, it is possible that the values of the dual variables λ and μ violate the constraints of Problem **P3**. By adopting the dual fitting techniques, we can find a good fitting method to ensure the feasibilities of dual variables.

Lemma 4: The values of dual variables in the $r-1$th round are denoted by $(\lambda_{r-1},\mu_{r-1})$. It ensures that $(\lambda_{r-1},\delta h(\mu_{r-1},c_r^*)\mu_{r-1})$ is feasible to the dual problem **P4**, where $h(\mu_{r-1},c_r^*)=(\tilde{u}_{c_r^*}(D_{c_r^*}^*(t))/\omega_{c_r^*}(D_{c_r^*}^*(t))\mu_{r-1})$, $\delta=\max_{D_c'(t),\,D_c''(t)\in\mathbf{D}_c(t),\,c\in\mathcal{C}}(\omega_c(D_c'(t))/\omega_c(D_c''(t)))$, and c_r^* represents the chosen camera in the rth round iteration.

Proof: In line (7) of Algorithm 1, we can observe that λ_{c^*} is updated to $\tilde{u}_{c^*}(D_{c^*}^*(t))$. And $\lambda_{c^*}\geq\tilde{u}_{c^*}(D_{c^*}(t))$, $\forall D_{c^*}(t)\in\mathbf{D}_{c^*}(t)\setminus D_{c^*}^*(t)$. Thus, the first constraint of **P4** is satisfied for the cameras in the set $\mathcal{Q}$. As for the camera c in the set $\mathcal{C}\setminus\mathcal{Q}$, we have

$$\begin{aligned}h\left(\mu_{r-1},c_r^*\right)&=\frac{\tilde{u}_{c_r^*}\left(D_{c_r^*}^*(t)\right)}{\omega_{c_r^*}\left(D_{c_r^*}^*(t)\right)\mu_{r-1}}\\&\geq\frac{\tilde{u}_c\left(D_c^*(t)\right)}{\omega_c\left(D_c^*(t)\right)\mu_{r-1}}\quad\forall c\in\mathcal{C}\setminus\mathcal{Q}.\end{aligned}$$

Thus, it is true that

$$h\left(\mu_{r-1},c_r^*\right)\omega_c\left(D_c^*(t)\right)\mu_{r-1}\geq\tilde{u}_c\left(D_c^*(t)\right).$$

From the definition of δ, we have

$$\delta\geq\frac{\omega_c\left(D_c'(t)\right)}{\omega_c\left(D_c''(t)\right)}\geq 1.$$

Therefore, we can obtain the following expression:

$$\begin{aligned}\delta h\left(\mu_{r-1},c_r^*\right)\omega_c\left(D_c^*(t)\right)\mu_{r-1}\geq\delta\tilde{u}_c\left(D_c^*(t)\right)\geq\tilde{u}_c(D_c(t))\\\forall c\in\mathcal{C}\setminus\mathcal{Q},D_c(t)\in\mathbf{D}_c(t).\end{aligned}$$

Therefore, $(\lambda_{r-1},\delta h(\mu_{r-1},c_r^*)\mu_{r-1})$ is feasible to the optimization problem **P4**. ■

The selection process of Algorithm 1 and the dual fitting process determine the approximation ratio of Algorithm 1. We will continue to investigate the impacts of the above two processes by the following theorem.

Theorem 1: The approximation ratio of Algorithm 1 is $\eta=1+\delta(\Delta/\Delta-1)(e-1)$.

Proof: Denote the values of dual variables in the rth iteration by (λ^r,μ^r). And the last iteration of Algorithm 1 is denoted by R. We will derive the approximation ratio when Algorithm 1 stops under different conditions.

If the stopping condition is $\mathcal{Q}=\mathcal{C}$ and $\mu<\exp(\Delta-1)$. Lemma 3 proves the feasibility of the solution to Problem **P3**. The dual fitting techniques mentioned in Lemma 4 also ensure the feasibility to Problem **P4**. The weak duality of linear programming relaxation indicates that any feasible solution to the dual problem **P3** is an upper bound of Problem **P4**. Thus, the solution obtained by Algorithm 1 is an optimal solution to Problem **P3**, which indicates that the approximation ratio is 1.

If the stopping condition is $\mu_r\geq\exp(\Delta-1)$, we consider the case that satisfies $\exists r\leq R,\eta<(y/\sum_{c\in\mathcal{C}}\lambda_c^r)$, where y denotes the optimal result of the Problem **P4**. Since we have $\sum_{c\in\mathcal{C}}\lambda_c^r=u^r(t)$ and the nondecreasing property of $u^r(t)$, the approximation ratio $(y/\sum_{c\in\mathcal{C}}\lambda_c^r)$ must be less than η when the iteration continues. If for all $r\leq R$, $\eta<(y/\sum_{c\in\mathcal{C}}\lambda_c^r)$, then

$$\begin{aligned}\mu^r&=(\exp(\Delta-1))^{\omega_{c^*}\left(D_{c^*}^*(t)\right)/(B\tau-\Gamma)}\cdot\mu^{r-1}\\&=[(\exp(\Delta-1))^{\frac{\Gamma}{B\tau-\Gamma}}]^{\omega_{c^*}\left(D_{c^*}^*(t)\right)/\Gamma}\cdot\mu^{r-1}\\&=\left[1+\frac{\Phi}{\frac{B\tau}{\Gamma}-1}\right]^{\omega_{c^*}\left(D_{c^*}^*(t)\right)/\Gamma}\cdot\mu^{r-1}\end{aligned}$$

where $\Phi=((B\tau/\Gamma)-1)[(\exp(\Delta-1))^{(1/(B\tau/\Gamma)-1)}-1]$. Since $\Gamma\geq\omega_{c^*}(D_{c^*}(t)),\forall D_{c^*}(t)\in\mathbf{D}_{c^*}(t)$, and $(1+a)^x\leq 1+ax,\forall x\in[0,1]$, we have

$$\left[1+\frac{\Phi}{\frac{B\tau}{\Gamma}-1}\right]^{\omega_{c^*}\left(D_{c^*}^*(t)\right)/\Gamma}\leq 1+\frac{\Phi\omega_{c^*}\left(D_{c^*}^*(t)\right)}{B\tau-\Gamma}.$$

Moreover, since $\Delta=(B\tau/\Gamma)$, we have

$$\begin{aligned}B\tau\mu^r&\leq B\tau\cdot\left(1+\frac{\Phi\omega_{c^*}\left(D_{c^*}^*(t)\right)}{B\tau-\Gamma}\right)\cdot\mu^{r-1}\\&=B\tau\mu^{r-1}+\frac{B\tau}{B\tau-\Gamma}\Phi\omega_{c^*}\left(D_{c^*}^*(t)\right)\mu^{r-1}\\&=B\tau\mu^{r-1}+\frac{B\tau}{B\tau-\Gamma}\left(\frac{B\tau}{\Gamma}-1\right)\\&\quad\times\left[(\exp(\Delta-1))^{\frac{1}{\frac{B\tau}{\Gamma}}-1}-1\right]\\&=B\tau\mu^{r-1}\frac{\Delta}{\Delta-1}(\Delta-1)\\&\quad\times\left[(\exp(\Delta-1))^{\frac{1}{\Delta-1}-1}\right]\omega_{c^*}\left(D_{c^*}^*(t)\right)\mu^{r-1}\\&=B\tau\mu^{r-1}+\Delta(e-1)\omega_{c^*}\left(D_{c^*}^*(t)\right)\mu^{r-1}.\end{aligned}$$

From the definition of the function $h(\mu^{r-1}, c_r^*)$, we have

$$\omega_{c_r^*}\left(D_{c_r^*}^*(t)\right)\mu^{r-1} = \tilde{u}_{c^*}\left(D_{c_r^*}^*(t)\right)/h\left(\mu^{r-1}, c_r^*\right).$$

Note that $u^r(t) - u^{r-1}(t) = \tilde{u}_{c_r^*}(D_{c_r^*}^*(t))$; thus, the update of $B\tau\mu$ satisfies the following condition:

$$B\tau\mu^r \leq B\tau\mu^{r-1} + \Delta(e-1)\frac{u^r(t) - u^{r-1}(t)}{h\left(\mu^{r-1}, c_r^*\right)}.$$

Considering the dual fitting process mentioned in Lemma 4, the dual variables $(\lambda^{r-1}, \delta h(\mu^{r-1}, c_r^*)\mu^{r-1})$ are feasible to the dual problem **P4**, we have

$$y \leq \sum_{c\in\mathcal{C}} \lambda_c + \delta h\left(\mu^{r-1}, c_r^*\right)B\tau\mu^{r-1}.$$

We can further derive the bound of the function $h(\mu^{r-1}, c_r^*)$ as follows:

$$h\left(\mu^{r-1}, c_r^*\right) \geq \frac{y - \sum_{c\in\mathcal{C}}\lambda_c}{\delta B\tau\mu^{r-1}}.$$

Given that the approximation ratio η satisfies $\eta < (y/\sum_{c\in\mathcal{C}}\lambda_c^r), \forall c \leq R$. Then

$$\frac{1}{h\left(\mu^{r-1}, c_r^*\right)} \leq \delta\frac{B\tau\mu^{r-1}}{y - \sum_{c\in\mathcal{C}}\lambda_c} \leq \delta\frac{\eta}{\eta-1}\frac{B\tau\mu^{r-1}}{y}.$$

The approximation ratio after R iterations can be calculated as follows:

$$\begin{aligned} B\tau\mu^R &\leq B\tau\mu^{R-1} \\ &\quad + \delta\Delta(e-1)(u^R(t) - u^{R-1}(t))\frac{\eta}{\eta-1}\frac{B\tau\mu^{R-1}}{y} \\ &= B\tau\mu^{R-1}\left(1 + \delta\Delta(e-1)\frac{\eta}{\eta-1}\frac{u^R(t) - u^{R-1}(t)}{y}\right) \\ &\leq B\tau\mu^0\exp\left(\delta\Delta(e-1)\frac{\eta}{\eta-1}\frac{u^R(t)}{y}\right) \end{aligned}$$

where the last inequality is evaluated via the fact $1 + x \leq e^x, \forall x \geq 0$. From the stopping condition $\mu^R \geq \exp(\Delta - 1)$ and $\mu^0 = 1$, we can have

$$\exp(\Delta - 1) \leq \exp\left(\delta\Delta(e-1)\frac{\eta}{\eta-1}\frac{u^R(t)}{y}\right).$$

Thus

$$\frac{y}{u^R(t)} \leq \delta\frac{\Delta(e-1)}{\Delta-1}\frac{\eta}{\eta-1}.$$

As the solution obtained by Algorithm 1 is feasible to Problems **P3** and **P4**, the weak duality of linear programming relaxation indicates that $(y/u^R(t))$ is the upper bound of the approximation. Therefore

$$\begin{aligned} \delta\frac{\Delta(e-1)}{\Delta-1}\frac{\eta}{\eta-1} &\leq 1 + \delta\frac{\Delta(e-1)}{\Delta-1}\frac{1}{\eta-1} \\ &\leq 1 + \delta\frac{\Delta(e-1)}{\Delta-1} = \eta. \end{aligned}$$

Thus, the approximation ratio of Algorithm 1 is η, which proves the theorem. ■

Algorithm 2 Online Approximation Algorithm for Bandwidth Resource Allocation

Input:

Same as that in Algorithm 1.

Output:

Transmission decision $D_c^*(t), \forall c \in \mathcal{C}, t$, bandwidth allocation decision $b_c^*(t), \forall c \in \mathcal{C}, t$.

1: Initialization step: $t = 0$, $H_c(0) = 0, \forall c \in \mathcal{C}$.
2: **while** CVR service is active **do**
3: Calculate the transmission decisions $\{D_c^*(t), \forall c \in \mathcal{C}\}$ according to Algorithm 1.
4: Calculate the bandwidth allocation decisions $\{b_c^*(t), \forall c \in \mathcal{C}\}$ according to the transmission decisions $\{D_c^*(t), \forall c \in \mathcal{C}\}$.
5: Update each queue according to $H_c(t+1) = \max[H_c(t) - u_c(t) + u_0, 0]$.
6: Set $t \leftarrow t + 1$.
7: **end while**

B. Online Approximation Algorithm

After utilizing the Lyapunov decomposition techniques, the approximate online resource allocation algorithm can be realized through solving the one-shot problem **P2** in each time slot. However, the computational infeasibility of the one-shot problem motivates us to exploit Algorithm 1 to obtain a good-enough solution to the one-shot problem. By integrating the Lyapunov optimization algorithm and Algorithm 1, we obtain the final design of the online bandwidth resource allocation algorithm shown in Algorithm 2.

For line (4) of Algorithm 2, given the transmission decision $D_c^*(t)$, the allocated bandwidth for a camera c is calculated by $b_c^*(t) = (m_t^{c,b}D_c^b(t) + \sum_{e\in\mathcal{E}} m_t^{c,e}D_c^e(t)) \cdot l/\tau$.

Theorem 2: Denoting the optimal time-average overall utilities by $\bar{u}^*$, then the solution obtained by Algorithm 2 satisfies the following condition:

$$\lim_{T\to\infty}\sum_{c\in\mathcal{C}}\bar{u}_c(T) \geq \eta\bar{u}^* - \frac{\Lambda}{V}$$

where $\Lambda = N(u_0 - u_{\max})^2$, and η denotes the approximation ratio of Algorithm 1.

Proof: The Lyapunov drift-plus-penalty function is defined as follows:

$$\begin{aligned} \Delta(L(t)) - V\sum_{c\in\mathcal{C}}u_c(t) &\leq \sum_{c\in\mathcal{C}}[(u_0 - u_c(t))H_c(t)|\mathbf{H}(t)] \\ &\quad + N(u_0 - u_{\max})^2 - V\sum_{c\in\mathcal{C}}u_c(t). \end{aligned}$$

If the distribution of the utility function can be known beforehand, we can design a stationary randomized algorithm which purely depends on the probability distribution and achieves the optimal value $\bar{u}^*$. Moreover, it has been proven that any online algorithm cannot improve the solution of the stationary randomized algorithm.

Denoting $f(t) = \sum_{c\in\mathcal{C}}[(u_0 - u_c(t))H_c(t)|\mathbf{H}(t)] - V\sum_{c\in\mathcal{C}}u_c(t)$, we define the optimal value of Problem **P3** as $\bar{f}(t)$, and the optimal value to Problem **P2** as $f^*(t)$. It is true

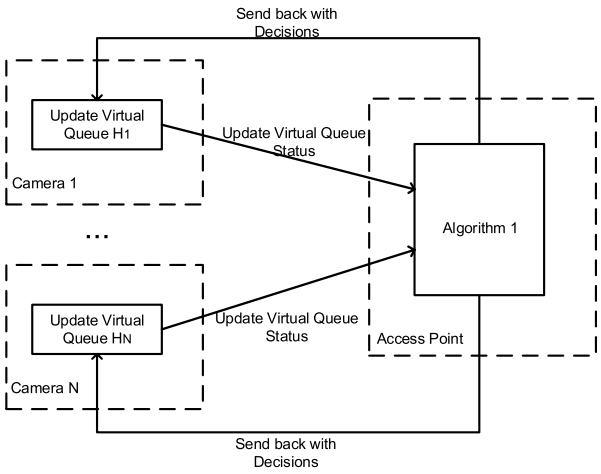


Fig. 3. Illustration on the implementation of Algorithm 2.

that $\bar{f}(t) = -f^*(t)$. From the approximation of Algorithm 1, we know that the value of $f(t) \leq -\eta \bar{f}(t) = \eta f^*$. Since the stationary randomized algorithm can satisfy that $u_0 \leq \bar{u}_c^*$, we have

$$\Delta(L(t)) - V \sum_{c \in \mathcal{C}} u_c(t) \leq \Lambda + \eta \sum_{c \in \mathcal{C}} \left[\left(u_0 - \bar{u}_c^* \right) H_c(t) | \mathbf{H}(t) \right] \\ -\eta V \bar{u}^* \leq \Lambda - \eta V \bar{u}^*$$

where $\Lambda = N(u_0 - u_{\max})^2$. Thus

$$\Delta(L(t)) \leq \Lambda - \eta V \bar{u}^* + V \sum_{c \in \mathcal{C}} u_c(t).$$

We can further obtain

$$L(\mathbf{H}(T)) - L(\mathbf{H}(1)) \leq T\Lambda - T\eta V \bar{u}^* + V \sum_{t=1}^{T} \sum_{c \in \mathcal{C}} u_c(t).$$

Dividing both sides of the above inequality, we have

$$\frac{L(\mathbf{H}(t))}{T} - \frac{L(\mathbf{H}(1))}{T} \leq \Lambda - \eta V \bar{u}^* + V \frac{1}{T} \sum_{t=1}^{T} \sum_{c \in \mathcal{C}} u_c(t).$$

Taking the limitation $T \to \infty$, we have

$$\lim_{T \to \infty} \sum_{c \in \mathcal{C}} \bar{u}_c(t) \geq \eta \bar{u}^* - \frac{\Lambda}{V}$$

which proves the theorem. ■

Theorem 2 indicates that the distance between the solution achieved by Algorithm 2 and the η-approximate optimal solution can be controlled by the parameter V within any small distance.

From the description of Algorithm 2, we can observe that our strategy can be divided into two separate components. An illustrative example on how to implement Algorithm 2 is shown in Fig. 3. Each camera needs to maintain its own virtual queue, and sends the updated status of virtual queues to the controller implemented in the AP. After receiving the status of virtual queues from all cameras, the controller will run Algorithm 1 to calculate the resource allocation decisions for the next time slot. Then, the controller will send decisions back to all cameras, and cameras can adapt the video playback rate according to the resource allocation decisions. Moreover, there are several possible approaches to implement our proposed algorithm in real CVR systems. In our system model, we consider the case that the AP can be customized by the designer (e.g., using OpenWRT [37]). The concept of media cloud [29], [30] indicates that it is also feasible to implement the algorithm on the edge server. For different approaches, it is important to take various factors, such as the access to physical hardware, the difficulty in customizing hardware, into account during algorithm implementation.

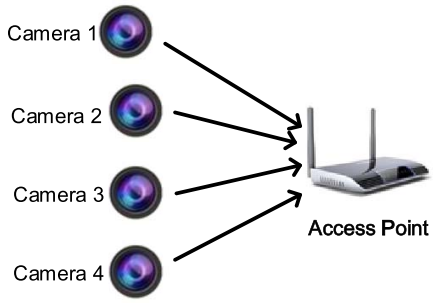


Fig. 4. Simulation scenario with four wireless cameras and one AP.

V. Performance Evaluation

In this section, we conduct a set of real-trace driven simulations to evaluate the effectiveness of our proposed algorithm.

A. Evaluation Settings

In our experiments, the upstream bandwidth capacity B of the AP is set as 4 Mb/s, which is less than the amount to transmit all layers of all video streams simultaneously. There are four wireless cameras in the monitored region and the simulation scenario can be shown in Fig. 4. For each video stream, there are three layers whose playback rates are 0.5, 1, and 2 Mb/s, respectively. Note that, the playback rate of a video layer means the amount of bandwidth resource needed to transmit video packets up to this layer. The value of V is set to be 10, the number of time slots is 450, and the length of one time slot is 1 s.

As the utility of a CVR user is affected by multiple factors, such as streaming rate and video content characteristics. We consider three types of utility function in our simulation.

1) *Rate-Based Utility Function (RUF):* The utility received by a CVR user increases concavely with the streaming rate [7], [35]. Specifically, the utility is defined as a function of the number of video layers transmitted from the camera. That is, the utility function $u_c(t)$ is defined as follows:

$$u_c(t) = \log\left(1 + D_c^b(t) + \sum_{e \in \mathcal{E}} D_c^e(t)\right).$$

2) *Content-Based Utility Function (CUF):* Video content in each video frame determines the information that a CVR user can obtain from that frame [34]. We use the number of moving objects in a video frame to represent the

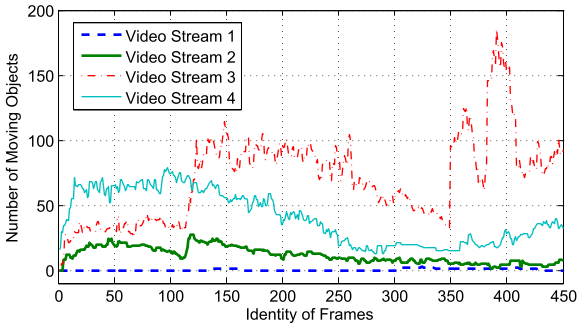


Fig. 5. Number of moving objects in each video frame.

volume of video content. Denote the number of moving objects in the video frame f_t^c by o_t^c, then the utility function is given as follows:

$$u_c(t) = \upsilon_b D_c^b(t) \log\left(o_t^c\right) + \sum_{e \in \mathcal{E}} \upsilon_e D_c^e(t) \log\left(o_t^c\right)$$

where υ is the importance priority of a video layer.

3) *Weighted Utility Function (WUF):* It is a linear weighted sum of RUF and CUF.

At first, we evaluate the bandwidth resource allocation strategies under RUF and CUF, respectively. When considering RUF, one video stream consists of one base layer and two enhancement layers, which correspond to the playback rate 0.5, 1, and 2 Mb/s. When specifying the utility as CUF, we analyze the number of moving objects in four different video streams, which is shown in Fig. 5. The four video streams represent different scenarios, including nearly static and highly active scenarios. Note that we have used 20 representative video streams in total from a CVR service provider, and all of our experiment results are based on these real video traces.

We denote the bandwidth resource allocation strategy illustrated by Algorithm 2 as CRA cooperative resource allocation (CRA) algorithm, and also implement several other algorithms in our simulations for the sake of comparison.

1) *Static Resource Allocation (SRA):* The amount of bandwidth resource allocated to each video stream is determined at the beginning of transmission and the allocation will keep unchanged.
2) *Least Required Resource Allocation (LRA):* At each time slot, the transmission of the base-layer video content is ensured for each video stream. The unallocated upstream bandwidth will be allocated among video streams evenly.

B. Simulation Results

Fig. 6 shows the allocation strategies on upstream bandwidth for transmitting the first 100 frames under different types of utility functions. From Fig. 6(a), we can observe that upstream bandwidth is allocated to each video stream in a time-divided manner. However, when defining utility function based on the number of moving objects, more upstream bandwidth will be allocated to video streams with more moving objects, such as video stream 4, which has the largest number of moving objects across the first 100 frames shown

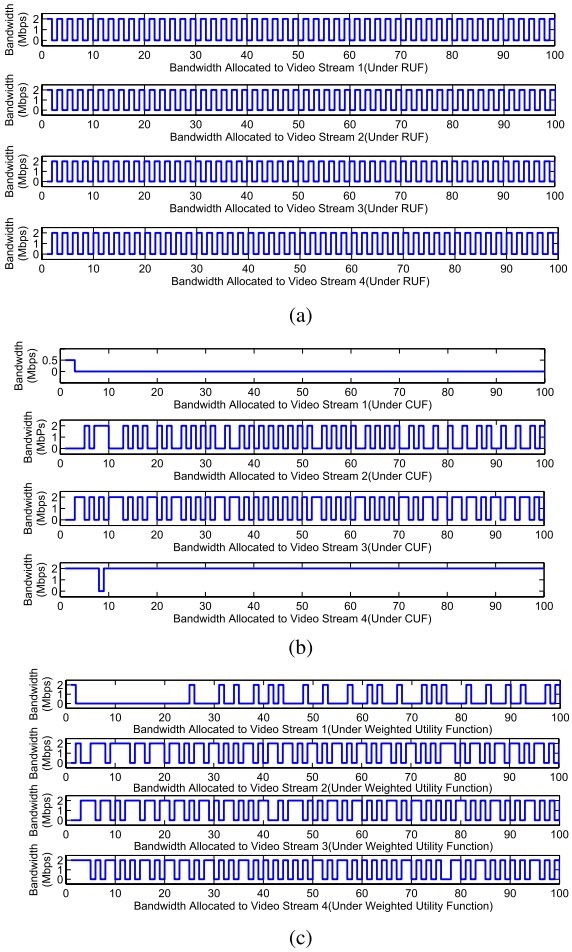


Fig. 6. Upstream bandwidth allocation under different types of utility functions. (a) Bandwidth allocation under RUF. (b) Bandwidth allocation under CUF. (c) Bandwidth allocation under WUF.

in Fig. 6(b). In this case, video streams with few moving objects can hardly obtain bandwidth resource.

The results shown in Fig. 6 show that CRA allocates upstream bandwidth in an ON–OFF pattern due to the time-average constraint (e) in **P1**. If we need to force CRA to guarantee that the utility for each video stream at each time slot is higher than the basic utility requirement u_0 (that is $u_c(t) \geq u_0, \forall t, c$), we can simply allocate the basic required bandwidth for each video stream before running CRA at each time slot. CRA will determine how to allocate the left bandwidth resource.

Fig. 6(c) shows that the first video stream can be allocated with more bandwidth than the case in Fig. 6(a) under WUF when the weight is set to 0.5. Moreover, we illustrate the time-average user-received utility when tuning the weight from 0 to 1 in Fig. 7. A larger α results in allocating upstream bandwidth periodically, while a smaller α puts more weights on the factor of the number of moving objects. Under the above definition of utility functions, when transmitting video frames with the highest rate, the value of CUF can be higher than RUF if the number of moving objects is large. From Fig. 6(c), we can observe that the first video stream can also be allocated with upstream bandwidth even if it has

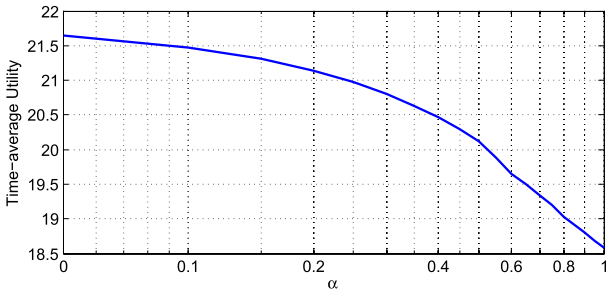


Fig. 7. Time-average weighted utility when varying the weight.

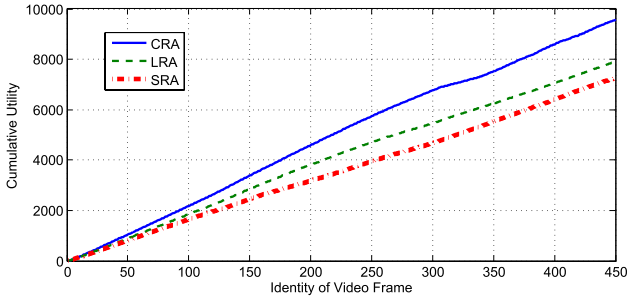


Fig. 8. Cumulative utility under different bandwidth resource allocation strategies.

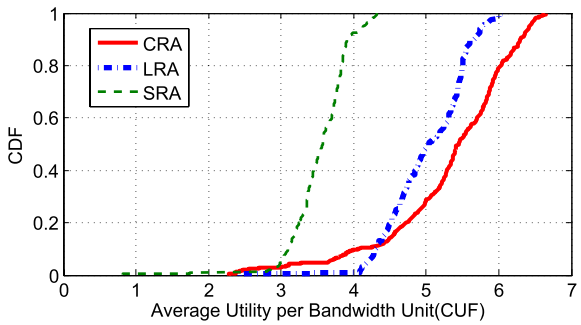


Fig. 9. Average utility per bandwidth unit under different bandwidth resource allocation strategies.

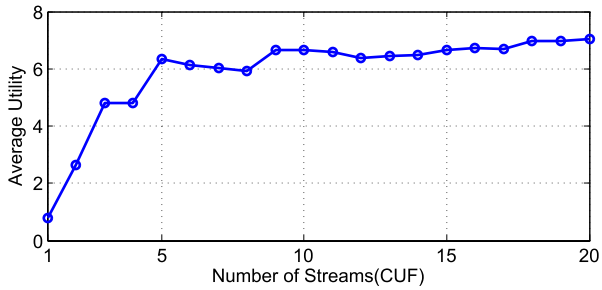


Fig. 10. Average utility with various numbers of streams under CUF.

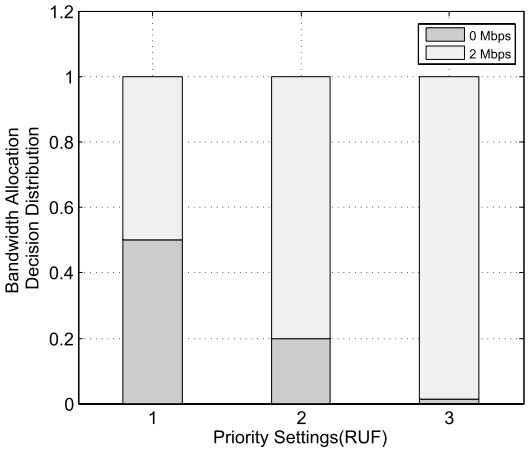
almost no moving objects. However, the amount of upstream bandwidth allocated to video stream 1 is much less than that under RUF. Since the bandwidth value of zero indicates that the corresponding video frames will not be available. Thus, the weighted utility function is an efficient way to balance video frame availability and utility.

If the utility function is defined based on streaming rate, then the user-received utility of each frame is deterministic under SRA and LRA. Thus, we analyze the user utility defined by CUF. Fig. 8 shows the cumulative utility under different bandwidth resource allocation strategies. The SRA strategy allocates the upstream bandwidth resource evenly to video streams. And the LRA strategy allocates sufficient bandwidth to each video stream to ensure that user-received utility can exceed the utility threshold u_0. From Fig. 8, we can see that the proposed CRA strategy can increase the user-received utility by 21% and 32% compared with that under LRA and SRA, respectively.

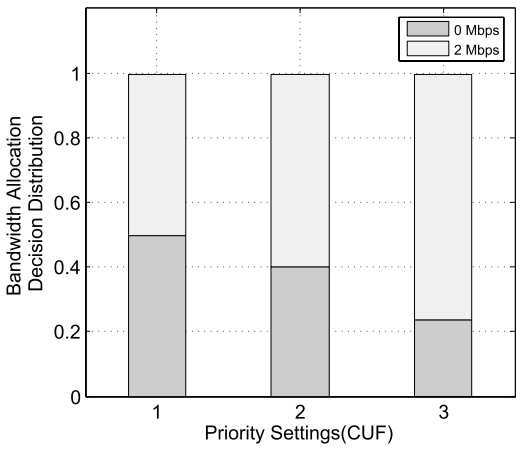
To maximize the overall utility, it is desirable to fully utilize each bandwidth unit as much as possible. We define a metric called *average utility per bandwidth unit* (e.g., 1 kb/s) to verify how well each resource allocation strategy achieves the goal. The measure of the average utility per bandwidth unit is calculated by dividing the utility of each video frame by the bandwidth allocated for transmitting that frame.

Fig. 9 shows the CDF of average utility across 450 video frames. We only conduct the experiments under CUF since all resource allocation strategies achieve the constant average utility per bandwidth unit under RUF. The SRA strategy can achieve the lowest average utility since its decisions are independent of the number of objects in video frames. Although the CRA results in that 10% of bandwidth has an average utility value lower than 4, it can also make around 20% of bandwidth have an average utility value higher than 6. Under LRA, all bandwidth will at least have an average utility value of 4. Since we do not exploit prediction technologies, the greedy allocation of CRA will not divide the bandwidth into different streams even though all of them have objects. The CRA strategy makes decisions based on the number of objects in several past frames instead of the instantaneous video frame. If the video streams have a similar number of objects in some frames, LRA can obtain more utilities by dividing bandwidth evenly across all video streams. However, the CRA strategy can still ensure that around 20% of bandwidth has a higher average utility value than that under the LRA strategy. The diversities of the number of objects in different streams across video frames can guarantee that CRA achieves much higher utility than that of other strategies.

With the increasing number of video streams, the shortage of upstream bandwidth will be even worse. Instead of provisioning more bandwidth, it is expected that the resource allocation strategy can exploit the existing resource in a better way. To evaluate whether the CRA strategy is robust to the increasing number of video streams, we choose 20 video streams from our data traces, and randomly select several of them to feed into our experiments. When the number of streams is smaller than 5, we found that upstream bandwidth is not fully utilized and the average utility is lower than that in other cases. If we vary the number of video streams from 5 to 20, the average utility obtained in all of these cases remains around 6.5, as shown in Fig. 10. Therefore, even if the shortage of upstream bandwidth becomes more severe, the CRA strategy can still ensure upstream bandwidth



(a)



(b)

Fig. 11. Bandwidth allocation decision under different priority settings. (a) RUF. (b) CUF.

is allocated to video frames who have a high utility value. Note that if we classify streams into multiple classes based on the average number of moving objects (e.g., $<$10, 10–50, and $>$50 objects), the ratio between the maximal average utility and the minimal utility within each class is around 1.5. Considering the weighted utility function, the ratio can be reduced to around 1.2. Thus, the weighted utility function can help to improve fairness across video streams.

Our approach allows the differentiation of video streams according to user preference settings. In our simulations, we specify the priorities of video streams by assigning different baseline utility requirements c_0 to each camera. In the experiment, we set the priorities of the first three video streams to the same value, and set the priority of the fourth video stream to a higher value.

Fig. 11 shows the distribution of video frames of the fourth video stream, in which the AP allocates 0 or 2 Mb/s, respectively. The distribution of video frames can characterize the bandwidth allocation decision distribution. Due to the greedy allocation property of the CRA strategy, one video frame can only be allocated with either the highest transmission rate or no bandwidth. From Fig. 11, we can observe that the CRA strategy under RUF is more sensitive to priority variations than the CRA strategy under CUF. Specifically, when increasing the priority to the second video stream, the fraction of video frames being transmitted with 2 Mb/s under RUF is 20% larger than that under CUF. These results show that our CRA strategy can well adapt to different user preferences.

VI. Conclusion and Discussion

In this paper, we proposed an intelligent bandwidth allocation algorithm to achieve efficient upstream bandwidth multiplexing among distributed cameras in CVR services. After formulating the problem as a constrained stochastic optimization problem, we exploited the Lyapunov optimization framework to decompose the problem and explicitly proved their NP-hardness. To tackle the problem, we further proposed an approximation algorithm to solve the decomposed problems. By applying Lyapunov decomposition and integer approximation techniques, we designed a hierarchical approximation algorithm called CRA to reduce the high computation complexity and avoid the use of any prediction. We also explicitly proved the approximation ratio of our proposed CRA algorithm. By conducting extensive trace-driven simulations, we demonstrated that our proposed algorithm improves the user-received utility by more than 20%, and ensures the stability of the average utility per bandwidth unit even with the increasing number of video streams under constrained bandwidth capacity. Our algorithm can also take the priority of video streams into account according to user preference, and increase upstream bandwidth allocation to video streams with a higher priority. In our future work, we plan to implement our algorithm in a real CVR system and further explore the effectiveness of other types of utility functions.

References

[1] N. Haering, P. L. Venetianer, and A. Lipton, "The evolution of video surveillance: An overview," *Mach. Vis. Appl.*, vol. 19, nos. 5–6, pp. 279–290, 2008.

[2] C. Le Barz and T. Lamarque, "Video surveillance cameras," in *Intelligent Video Surveillance Systems*. Hoboken, NJ, USA: Wiley, Feb. 2013, pp. 47–64.

[3] *Dropcam*, accessed on Jul. 8, 2015. [Online]. Available: https://www.dropcam.com

[4] *iVideon*, accessed on Jul. 8, 2015. [Online]. Available: http://www.ivideon.com/

[5] *Cloud DVR*, accessed on Jul. 8, 2015. [Online]. Available: http://www.alcatel-lucent.com/solutions/cloud-dvr

[6] *Cloud Video Surveilance Report*, accessed on Jul. 8, 2015. [Online]. Available: http://www.eagleeyenetworks.com/cloud-video-surveillance-report-2014/

[7] J. Xu, Y. Andrepoulos, Y. Xiao, and M. van der Schaar, "Non-stationary resource allocation policies for delay-constrained video streaming: Application to video over Internet-of-Things-enabled networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 782–794, Apr. 2014.

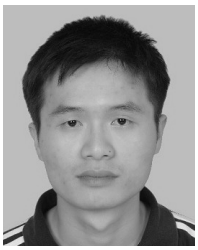
[8] A. Seema and M. Reisslein, "Towards efficient wireless video sensor networks: A survey of existing node architectures and proposal for a flexi-WVSNP design," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 462–486, Sep. 2011.

[9] S. Bouaziz, M. Fan, A. Lambert, T. Maurin, and R. Reynaud, "PICAR: Experimental platform for road tracking applications," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2003, pp. 495–499.

[10] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "MeshEye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *Proc. 6th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2007, pp. 360–369.

[11] J. Campbell, P. B. Gibbons, S. Nath, P. Pillai, S. Seshan, and R. Sukthankar, "IrisNet: An Internet-scale architecture for multimedia sensors," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 81–88.

[12] J. Polastre *et al.*, "A unifying link abstraction for wireless sensor networks," in *Proc. 3rd Int. Conf. Embedded Netw. Sensor Syst.*, 2005, pp. 76–89.
[13] S. Rein and M. Reisslein, "Low-memory wavelet transforms for wireless sensor networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 291–307, May 2011.
[14] T.-C. Wang, H.-C. Fang, and L.-G. Chen, "Low-delay and error-robust wireless video transmission for video communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1049–1058, Dec. 2002.
[15] C. Chen, R. W. Heath, Jr., A. C. Bovik, and G. de Veciana, "Adaptive policies for real-time video transmission: A Markov decision process framework," in *Proc. ICIP*, 2011, pp. 2249–2252.
[16] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 265–278, May 2012.
[17] 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.
[18] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 157–168.
[19] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: A QoE-aware dash system," in *Proc. 3rd Multimedia Syst. Conf.*, 2012, pp. 11–22.
[20] Z. G. Li *et al.*, "Adaptive rate control for H.264," *J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 376–406, 2006.
[21] J. Yang, X. Fang, and H. Xiong, "A joint rate control scheme for H.264 encoding of multiple video sequences," *IEEE Trans. Consum. Electron.*, vol. 51, no. 2, pp. 617–623, May 2005.
[22] Z. He and D. O. Wu, "Linear rate control and optimum statistical multiplexing for H.264 video broadcast," *IEEE Trans. Multimedia*, vol. 10, no. 7, pp. 1237–1249, Nov. 2008.
[23] S. Ren and M. van der Schaar, "Efficient resource provisioning and rate selection for stream mining in a community cloud," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 723–734, Jun. 2013.
[24] K. J. Ma, R. Bartos, S. Bhatia, and R. Nair, "Mobile video delivery with HTTP," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 166–175, Apr. 2011.
[25] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proc. 8th Int. Conf. Emerg. Netw. Experim. Technol.*, 2012, pp. 109–120.
[26] F. Molazem Tabrizi, J. Peters, and M. Hefeeda, "Dynamic control of receiver buffers in mobile video streaming systems," *IEEE Trans. Mobile Comput.*, vol. 12, no. 5, pp. 995–1008, May 2013.
[27] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 59–69, May 2011.
[28] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. M. Lau, "CloudMedia: When cloud on demand meets video on demand," in *Proc. IEEE 31st IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2011, pp. 268–277.
[29] Y. Wen, X. Zhu, J. J. P. C. Rodrigues, and C. W. Chen, "Cloud mobile media: Reflections and outlook," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 885–902, Jun. 2014.
[30] Y. Jin, Y. Wen, and C. Westphal, "Optimal transcoding and caching for adaptive streaming in media cloud: An analytical approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 1914–1925, Dec. 2015.
[31] 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.
[32] J. Lee and S. Bahk, "On the MDP-based cost minimization for video-on-demand services in a heterogeneous wireless network with multihomed terminals," *IEEE Trans. Mobile Comput.*, vol. 12, no. 9, pp. 1737–1749, Sep. 2013.
[33] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, "An evaluation of bitrate adaptation methods for HTTP live streaming," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 693–705, Apr. 2014.
[34] S. Tavakoli, J. Gutiérrez, and N. Garcia, "Subjective quality study of adaptive streaming of monoscopic and stereoscopic video," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 684–692, Apr. 2014.
[35] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 389–400.
[36] Z. He and D. Wu, "Resource allocation and performance analysis of wireless video sensors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 590–599, May 2006.
[37] *OpenWRT*, accessed on Jul. 8, 2015. [Online]. Available: https://openwrt.org/



Jian He received the B.S. and M.S. degrees from Sun Yat-sen University, Guangzhou, China, in 2011 and 2014, respectively.

His current research interests include content distribution networks, data center networking, green networking, and network measurement.

Di Wu (M'06) 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 Post-Doctoral 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 Associate Department Head of the Department of Computer Science with Sun Yat-sen University, Guangzhou, China. His current research interests include multimedia communication, cloud computing, peer-to-peer networking, Internet measurement, and network security.

Dr. 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.

Xueyan Xie received the B.S. degree in microelectronic from Fudan University, Shanghai, China, in 2013. He is currently pursuing the master's degree with the Department of Computer Science, Sun Yat-sen University, Guangzhou, China, under the supervision of Prof. D. Wu.

His current research interests include multimedia communication, content distribution networks, and cloud computing.

Min Chen (M'08–SM'09) received the B.S. degree in electronic and communication engineering from the South China University of Technology, in 1999, and the M.S and Ph.D. degrees in electronic and information technology from the South China University of Technology, in 2001 and 2004 respectively.

He was the Research and Development Director of Confederal Network Inc., Renton, WA, USA, from 2008 to 2009. He was an Assistant Professor with the School of Computer Science and Engineering, Seoul National University (SNU), Seoul, Korea, from 2009 to 2012. He was a Post-Doctoral Fellow with SNU for one and a half years, and the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, for three years. He is currently a Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He has authored over 180 paper publications, including 85 SCI papers. His current research interests include Internet of Things, big data, machine-to-machine communications, body area networks, e-healthcare, mobile cloud computing, ad hoc cloudlet, cloud-assisted mobile computing, ubiquitous network and services, and multimedia transmission over wireless network.

Mr. Chen received the best paper award from the IEEE International Conference on Communications (ICC) in 2012, and the Best Paper Runner-Up Award from QShine in 2008. He is the Chair of the IEEE Computer Society Special Technical Communities on Big Data. He serves as an Editor or Associate Editor of *Information Sciences*, *Wireless Communications and Mobile Computing*, *IET Communications*, *IET Networks*, the *International Journal of Security and Communication Networks* (Wiley), the *Journal of Internet Technology*, *KSII Transactions on Internet and Information Systems*, and the *International Journal of Sensor Networks*. He is the Managing Editor of the *International Journal of Autonomous and Adaptive Communications Systems* and the *International Journal of Advanced Research and Technology*. He is a Guest Editor of the *IEEE Network* and the *IEEE Wireless Communications Magazine*. He was the Co-Chair of the IEEE ICC 2012—Communications Theory Symposium and the IEEE ICC 2013—Wireless Networks Symposium. He was the General Co-Chair of the IEEE International Conference on Computer and Information Technology in 2012 and Mobimedia in 2015. He was the General Vice Chair of Tridentcom in 2014. He was a Keynote Speaker of CyberC and Mobiquitous in 2012. He was a TPC Member of the IEEE INFOCOM in 2013 and 2014.

Yong Li (M'09) received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree in electronics engineering from Tsinghua University, Beijing, China, in 2012.

He was a Visiting Research Associate with Telekom Innovation Laboratories, Berlin, Germany, and the Hong Kong University of Science and Technology, Hong Kong, in 2012 and 2013, respectively. From 2013 to 2014, he visited the University of Miami, Coral Gables, FL, USA, as a Visiting Scientist. He is currently a Faculty Member of Electronic Engineering with Tsinghua University. He has authored over 100 research papers and holds ten granted and pending Chinese and International patents. His current research interests include networking and communications.

Dr. Li's research is granted by the Young Scientist Fund of Natural Science Foundation of China, the Post-Doctoral Special Fund of China, and industry companies, such as Hitachi and ZET. He served as the Technical Program Committee (TPC) Chair of the WWW Workshop of Simplex in 2013. He served on the TPC of several international workshops and conferences. He is an Associate Editor of the *EURASIP Journal on Wireless Communications and Networking*.

Guoqing Zhang received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

He is currently a Principal Investigator with the Chinese Academy of Sciences. His current research interests include information networks and network science, in particular, topology analysis of the Internet, online social networks and big data, and topology-aware optimization and application.