This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

Online Cloud Transcoding and Distribution for Crowdsourced Live Game Video Streaming

Yuanhuan Zheng, Di Wu, *Member, IEEE,* Yihao Ke, Can Yang, *Member, IEEE,* Min Chen, *Senior Member, IEEE*, Guoqing Zhang

***Abstract*—In recent years, empowered by rich media generation devices and convenient Internet access, *Crowdsourced Live Game Video Streaming (CLGVS)* has become one of the most popular Internet services. Twitch.tv, the most well-known CLGVS platform in the world, allows gamers to broadcast their gaming videos over the Internet. With the prevalence of mobile devices, viewers can watch gamers playing video games anywhere, anytime, on any devices (e.g., smartphones, tablets, or personal computers). However, the heterogeneity of user devices makes conventional solutions hard to ensure user-perceived quality. In this paper, we address the problem of cost-effective adaptive live game video streaming from the perspective of CLGVS service providers. Our purpose is to minimize the operational cost for CLGVS service providers by making live transcoding decisions, bit-rate adaptation decisions and datacenter assignment decisions dynamically. Meanwhile, our algorithm also ensures good-enough service quality for viewers. Due to the diversity of game genres, we also take game genres into account when designing our algorithm. To achieve the above purpose, we formulate the problem into a constrained stochastic optimization problem. By leveraging the Lyapunov optimization framework, we derive the online strategy with provable performance bound. To evaluate the effectiveness of our proposed algorithm, we further conduct a series of trace-driven simulations. The experimental results demonstrate the effectiveness of our algorithm in terms of operational cost and service quality. Our proposed algorithm can reduce operational cost by up to 50% while achieving good-enough viewer QoE compared with other alternatives.**

***Index Terms*—video transcoding, cloud computing, video distribution, crowdsourced streaming, video games**

I. Introduction

The past decade has witnessed the explosive growth in live game video streaming. Twitch.tv [1] is one of the most successful live game video streaming service providers in the world. The amazing popularity has made Twitch.tv become the fourth largest source of the peak Internet traffic in the U.S. in February 2014 [2]. Twitch.tv currently attracts more than 1 million monthly active broadcasters and over 45 million monthly viewers [3]. The success of Twitch.tv proves the huge potential of the live game video streaming service, and has also attracted great attentions from both industry and academia.

The revolution of the mobile Internet driven by smart and powerful mobile devices has greatly enriched the sources of video platforms. Such change has created a new kind of video platform, called *crowdsourced live streaming platform* [4]. Crowdsourced live streaming platforms not only serve massive audience worldwide, but receive contents from many sources in the crowd. By combining crowdsourced technology and live game video streaming, Twitch.tv provides a new form of video service, namely crowdsourced live game video streaming (CLGVS) service.

Different from the traditional video-on-demand services, such as YouTube [5], Hulu [6] and so on, CLGVS platforms allow individual gamers to broadcast live game videos over the Internet. By leveraging the CLGVS service, gamers can stream game videos and communicate with viewers in real time. Meanwhile, viewers can make comments or even directly discuss with gamers via real-time chatting. The CLGVS service has already become a new type of video entertainment, and brought remarkable commercial interest. CLGVS platforms also support game video streams generated by various devices, such as personal computer, PS4, Xbox, and so on.

The existing CLGVS service providers (e.g. Twitch.tv [1], Douyu.tv [7], etc.) usually build their own private datacenters (with dedicated hardware) and private networks to ensure the service quality. To provide adaptive video streaming services, CLGVS service providers leverage their own private datacenters to transcode the live video streams into multiple versions and rely on CDNs (Content Distribution Networks) to deliver live video streams to geo-distributed viewers [4] [8]. However, such traditional architecture has the following shortcomings: *First*, it is costly for CLGVS service providers to build the private datacenters and private networks. According to the previous study [8], the number of online users may change dramatically over time. The traditional private datacenters can not provide elastic computation resources, and may incur a huge waste of money. *Second*, with conventional techniques, CLGVS service providers have to provide live video transcoding service themselves. But live video transcoding is computation-intensive and it is hard to provide elastic

Manuscript received December 17, 2015; revised February 26, 2016; accepted April 2, 2016. This work was supported in part by the National Science Foundation of China under Grant 61272397, Grant 61572538, Grant 61174152, Grant 61331008, Grant 61572220, in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant S20120011187. This paper was recommended by Associate Editor J. Liang. *(Corresponding author: Di Wu.)*

Y. Zheng, D. Wu and Y. Ke are with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China, Guangdong Province Key Laboratory of Big Data Analysis and Processing, Sun Yat-sen University, Guangzhou 510006, China, and also with Collaborative Innovation Center of High Performance Computing, National University of Defense Technology, Changsha 410073, China. (e-mail: zhyhuan@mail2.sysu.edu.cn, wudi27@mail.sysu.edu.cn, keyihao@mail2.sysu.edu.cn).

C. Yang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: cscyang@scut.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).

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

transcoding resources. *Third*, in the existing architectures, CLGVS service providers rely on CDNs to deliver live video streams. Although, it is convenient for service providers to build CLGVS service over CDNs, but it is hard for CLGVS service providers to customize the delivery strategy for every viewers to ensure the service quality.

The emerging technology of cloud computing releases video streaming service providers from building large, expensive private datacenters. By elastic resource provisioning, the cloud platform can offer a perfect solution for cost-effective video streaming service. Moreover, the cloud-based live transcoding services (e.g. Zencoder [9], Encoding.com [10], etc.), relieve the streaming platform from computation-intensive live transcoding via offering elastic transcoding resources. When receiving game video streams from gamers, the CLGVS platform exploits cloud transcoding services to transcode original streams to multiple versions and delivers them to viewers via geo-distributed datacenters. The CLGVS platform can also help to select a proper version of video streams according to the viewer's network conditions.

However, it is very difficult to design a cost-effective live game video streaming platform that can satisfy viewers with high service quality. The challenges are multifaceted: *First*, the heterogeneity of user devices (e.g. PCs, smartphones, tablets, etc.) demands multiple versions of original video streams. There exists significant diversity among user devices on their computational capacity, screen resolution, network bandwidth and so on. Therefore, CLGVS platforms should be able to provide the most appropriate video version. *Second*, compared with the conventional video-on-demand services, the video sources of a CLGVS platform are often generated by individual gamers. These video sources are much more dynamic than dedicated content providers, as gamers can start or terminate video streaming at their own will. In addition, the unpredictable wireless network conditions make it hard to ensure service quality. *Third*, the geo-distributed nature of viewers makes it hard to deliver high-quality video streams to each viewer. To guarantee the global accessibility of high quality streaming service, the CLGVS service provider should deploy its platform on multiple datacenters. *Fourth*, different game genres require various levels of video quality to achieve basic experience, which should therefore be taken into account for resource provisioning and bit-rate selection. *Fifth*, both live transcoding and video delivery incur huge monetary cost, including live transcoding cost, bandwidth cost and so on. To be successful in the market competition, CLGVS providers should minimize the operational cost as much as possible, while providing good-enough service quality to viewers.

In this paper, we try to address the above-mentioned challenges from the perspective of CLGVS providers. Our proposed algorithm aims at assisting the CLGVS service providers to offer a cost-effective adaptive CLGVS service, in which the operational cost can be minimized as much as possible, while still ensuring good-enough viewing experience. The algorithm design includes three major components: dynamic live transcoding decisions, adaptive bit-rate assignment and intelligent datacenter selection. Mathematically, we formulate the problem into a constrained stochastic optimization problem, and exploit the Lyapunov optimization framework [11] to derive the online algorithm. Our algorithm can optimize transcoding decisions, bit-rate assignment and datacenter selection jointly. Theoretically, we prove our proposed algorithm can approach the optimality with a tunable bound. In summary, we mainly make three contributions:

- We consider the optimization problem of providing cost-effective CLGVS service from the perspective of CLGVS service providers. Our proposed algorithm aims at reducing the operational cost while still ensuring good-enough service quality by jointly optimizing the live transcoding decisions, bit-rate assignment and datacenter selection. In addition, in order to extend the applicability of our algorithm , we take game genres into consideration when designing online algorithms.
- We formulate the problem into a constrained stochastic optimization problem. By exploiting the Lyapunov optimization theory, we design an online algorithm called *OCTAD*, which can make dynamic transcoding decisions for each gamer, and perform adaptive bit-rate assignment and datacenter selection for each viewer. Through making decisions dynamically, our algorithm can significantly reduce operational cost and ensure viewer QoE.
- To evaluate the effectiveness of our proposed algorithm *OCTAD*, we conduct extensive trace-driven simulations. We utilize the real trace to make our simulations more realistic. Our experimental results show that, *OCTAD* can cut down operational cost by up to 50% while achieving good-enough viewer QoE compared with other algorithms.

The rest of this paper is organized as follows. Section II reviews previous related work. The system model and formulation are described in Section III. In Section IV, we describe the design of our proposed online algorithm *OCTAD*. The simulation and performance evaluation of *OCTAD* are presented in Section V. Finally, Section VI concludes the paper and discusses the future work.

II. Related Work

Due to the development of video streaming technology and the flourish of gaming industry, live game video streaming has gained a lot of attention from both industry and academia.

Although CLGVS services are emerging in recent years, many measurement works have been conducted to understand the architecture, system performance and user behaviors of CLGVS platforms. Shea *et al.* [3] presented an initial experiment-based performance study, in which they studied the architecture of real-world gaming and streaming platforms. Pires *et al.* [12] investigated Twitch.tv. and pointed out the difference of traffic characteristics between YouTube Live and Twitch.tv. Smith *et al.* [13] studied the gaming community phenomenon in live game video streaming platforms and compared with some typical cloud gaming platforms (e.g. OnLive[14]). In their measurement work, they divided viewers into several communities and investigated the characteristics of each community. Besides, there are some other works to study virtual communities in Twitch.tv. Hamilton *et al.* [15]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

found that many stream communities formed around shared identities, and also described the processes through which stream communities were formed.

Live video transcoding is important for CLGVS services. Wang *et al.* [16] proposed to leverage idle CDN computation resources to jointly transcode and deliver videos. Wu *et al.* [17] proposed a collaborative strategy that leverages peer-to-peer technologies to distribute video transcoding tasks among peers. Cheng *et al.* [18] presented a framework for cloud-based video transcoding in the context of mobile video conference. Based on the above transcoding framework, they further introduced a prediction-based scheduling algorithm to optimize both the latency requirement and cloud utility cost. Jin *et al.* [19] exploited a three-way tradeoff between the caching, transcoding and bandwidth costs to minimize the total operational cost for on-demand video services providers. According to the observation that most of viewers terminate viewing sessions within 20% of their durations, Gao *et al.* [20] proposed a partial transcoding scheme for content management in the media cloud, in which some segments are pre-processed and stored in the cache, and the other segments are transcoded on the fly.

To satisfy the global distribution requirements of video streaming, current systems mostly rely on content distribution networks (CDNs) [21], peer-to-peer (P2P) [22], or hybrid solutions [23]. The cloud platform with elastic resource allocation capability has become an effective solution for large-scale global video delivery. He *et al.* [24] investigated the optimal deployment problem of cloud-assisted video distribution services and explore the best tradeoff between the operational cost and the user experience. Wu *et al.* [25] exploits a geo-distributed cloud to support large-scale social media streaming applications. Xu *et al.* [26] investigated the application architecture, video generation, adaptation schemes and delivery strategies of three popular video telephony systems, namely, Google+, iChat and Skype.

Our work differs from previous works in three aspects: *First*, although there exist some studies on adaptive bit-rate selection and transcoding decisions, it is infeasible to simply combine them to make joint decisions. Unlike previous works, we consider gamers, viewers and service providers as a whole and make decisions for three roles jointly. *Second*, our algorithm is specifically customized for the CLGVS systems. Since different game genres have various characteristics, we also take the diversity of game genres into account when we develop our system model and conduct algorithm design. It ensures our proposed algorithm can be applied to more general scenarios. *Third*, by exploiting the Lyapunov optimization theory, our proposed algorithm can reduce operational cost significantly and still ensure good-enough viewer QoE. Another benefit of our algorithm is that *OCTAD* does not require any future knowledge about user behaviors and network conditions. By tuning the Lyapunov parameters, our algorithm can approach the optimality with infinitely small distance.

III. System Model and Architecture

A. System Model

In this section we consider a typical CLGVS platform as illustrated in Figure 1. The mathematical notations used in problem formulation are summarized in Table I.

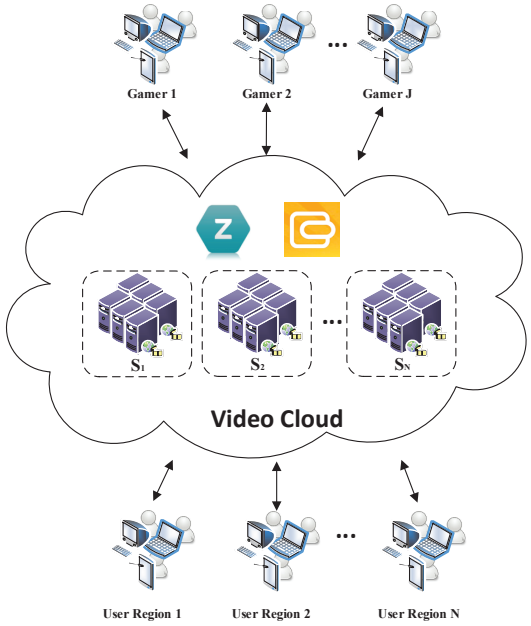


Fig. 1. A typical architecture of a crowdsourced live game video streaming (CLGVS) platform

CLGVS platforms (e.g., Twitch.tv [1], Douyu.tv [7]) allow gamers to broadcast their game videos over the Internet. The heterogeneity of user devices and the global distribution of viewer population make conventional solutions hard to ensure service quality. The emerging technology of cloud computing makes it much easier to solve these issues. In Figure 1, gamers stream their game videos to the CLGVS platform, which then delivers videos to viewers. In order to satisfy the requirements of viewers using different devices, the platform should transcode the original video streams to multiple versions and deliver the transcoded streams to viewers. In such an architecture, the service provider leverages cloud-based transcoding services, such as Zencoder [9], Encoding.com [10], Panda [27], to transcode the original video stream to various bit-rates. Moreover, geo-distributed datacenters of the cloud platform are used to deliver video streams to viewers in various regions.

Assume that a CLGVS service provider deploys streaming servers on N datacenters, denoted by $\{S_1, S_2, \cdots, S_N\}$. Whenever a user request arrives, the platform redirects it to a proper datacenter. We define $P(j,t)$ as the bit-rate of the video stream uploaded by gamer j at time slot t. In this model, we further assume the cloud transcoding service can transcode the original video stream to M different target bit-rates. Define $B_s(m)$ as a function to return the m-th target bit-rate, $m \in \{1, 2, \cdots M\}$. In our model, we consider a time-slotted system in which time is divided into a series of time slots and each time slot lasts for a period of τ. We further

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

suppose that the CLGVS service provider makes decisions at the beginning of each time slot. To facilitate our presentation, we define $B_v(i,t)$ as a function to obtain the bit-rate assigned to viewer i. We define $Z(i,t)$ as a function to obtain the index of the datacenter which viewer i connects to.

TABLE I
KEY MATHEMATICAL NOTATIONS IN SYSTEM MODEL

Notation	Description
N	The number of datacenters.
M	The number of target versions provided by cloud transcoding service.
τ	Length of time slot.
$P(j,t)$	The bit-rate of video stream uploaded by gamer j at time slot t.
$B_s(m)$	The m-th target bit-rate.
$B_v(i,t)$	The bit-rate assigned to viewer i at time slot t.
$Z(i,t)$	The index of the datacenter which viewer i connects to at time slot t.
$R(j,t)$	The set of indexes of bit-rates transcoded from gamer j's video stream at time slot t.
$G(i)$	The category that viewer i belongs to.
$E(k)$	The basic bit-rate required by viewers in the k-th category.
$U(n,t)$	The unit bandwidth price in datacenter n.
$W(n,t)$	The amount of bandwidth usage in datacenter n.
$I^t(i,n)$	Indicator that represents whether viewer i connects to datacenter n at time slot t.
$C_b(t)$	The bandwidth cost at time slot t.
$C_s(t)$	The transcoding cost at time slot t.
$C_o(t)$	The operational cost at time slot t.
$D_s(i,t)$	The transcoding delay experienced by viewer i at time slot t.
$D_n(i,t)$	The network delay experienced by viewer i at time slot t.
$D(i,t)$	The service delay experienced by viewer i at time slot t.
$Q(i,t)$	The QoE of viewer i at time slot t.

Define $R(j,t)$ as the set of indexes of bit-rates transcoded from gamer j's video stream in time slot t. To avoid useless transcoding operations (e.g., transcoding from a lower-bit-rate version to a higher-bit-rate version), we must ensure that:

$$P(j,t) \geq \max\{B_s(m), m \in R(j,t)\}, \forall j,$$

where $R(j,t) \subseteq \{1,2,\cdots,M\}$.

The CLGVS platform allows gamers to broadcast game videos in various categories. Each game genre has different requirements on video quality. According to the genre of game videos watched by viewers, we divide viewers into K categories, denoted as $G = \{G_1, G_2, \cdots, G_K\}$. Viewers in the same category have similar QoE requirements. We define a function $G(i)$ to obtain the category that viewer i belongs to. Let $E(k)$ be the basic bit-rate required by viewers in the k-th category. To ensure the basic QoE of each viewer, all viewers should receive a video stream with a bit-rate no less than $E(G(i))$. That means, in each time slot t, we must guarantee that:

$$B_v(i,t) \geq E(G(i)), \forall i,$$

where $G(i)$ returns the category that viewer i belongs to and $E(G(i))$ is the basic bit-rate that viewer i requires to get a just-good-enough experience, and the left-hand-side of the inequality is the bit-rate received by viewer i in time slot t.

B. Operational Cost Model

For geo-distributed datacenters, the bandwidth price varies across different regions. In time slot t, let $U(n,t)$ and $W(n,t)$ be the unit bandwidth price and the amount of bandwidth usage in datacenter n respectively. Assume the bandwidth price keeps constant within a time slot. Then the bandwidth cost of all datacenters can be given by:

$$C_b(t) = \sum_{n=1}^{N} U(n,t) \cdot W(n,t),$$

where $W(n,t) = \sum_{i \in V^t} B_v(i,t) \cdot I^t(i,n)$ and V^t be the set of viewers at the beginning of time slot t. $I^t(i,n)$ is an indicator that represents whether viewer i connects to datacenter n at time slot t. That is:

$$I^t(i,n) = 1 \ \{\text{viewer } i \text{ connects to datacenter } n\} \ .$$

We also define $C_b(i,t)$ as the bandwidth cost incurred by viewer i at time slot t. Then the bandwidth cost can be written as:

$$C_b(t) = \sum_{i \in V^t} C_b(i,t),$$

where $C_b(i,t) = B_v(i,t) \cdot U(Z(i,t),t)$.

In addition to the bandwidth cost, the CLGVS service provider also needs to pay for the transcoding cost. According to the current pricing model (e.g., Zencoder), the transcoding cost is closely related to the input bit-rate, target bit-rate and the video length. We define the transcoding cost incurred by gamer j at time slot t as below:

$$C_s(j,t) = \sum_{m \in R(j,t)} \widetilde{C}_s(P(j,t), B_s(m), t, \tau),$$

where $\widetilde{C}_s(P(j,t), B_s(m), t, \tau)$ is the service cost incurred by transcoding a video stream from a bit-rate $P(j,t)$ to a bit-rate $B_s(m)$ and τ is the length of a video. Let H^t be the online gamer set at the beginning of time slot t. Then the overall transcoding cost for all gamers can be represented as:

$$C_s(t) = \sum_{j \in H^t} C_s(j,t).$$

Transcoding cost and bandwidth cost make up most of the operational cost of CLGVS service providers for live game broadcasting. Compared with the above two kinds of cost, the fraction of other cloud service cost (e.g. storage cost, computation cost, etc.) is negligible. To simplify the problem formulation, we mainly consider transcoding cost and bandwidth cost in this paper. Therefore, in our model, the operational cost is defined as the sum of bandwidth cost and transcoding cost. Then the operational cost $C_o(t)$ in time slot t can be expressed as:

$$C_o(t) = C_b(t) + C_s(t).$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

C. Delay Model

According to [28], the delay constraint is one of the main challenges that a CLGVS service provider should face with. Similar to [29], we divide the delay experienced by a viewer into three components: *network delay* (at the network side), *transcoding delay* (at the server side) and *playout delay* (at the client side). Network delay is usually referred to as the *Round-Trip Time* (RTT), which can be measured by tools such as Ping and King [30]. Under the CLGVS scenario, network delay consists of three parts: gamer-side network delay, viewer-side network delay and transcoder-side network delay. Since gamer-side network delay and transcoder-side network delay won't be influenced by our decisions, we just take viewer-side network delay into consideration. Viewer-side network delay may be affected by multiple factors (e.g. network congestion, flash crowd, etc.). In case that these issues happen, the viewer-side network delay will increase significantly which will result in negative effects on the viewers' QoE. Transcoding delay is the difference between the time when the transcoding cloud receives the original stream and the time when the target versions are created. As for the playout delay, it is the time required for the user device to decode and display the video on the screen. Since the playout delay is usually constant and not affected by the server side, we do not take it into account for brevity. Thus, we re-define the total delay experienced by viewer i as the sum of transcoding delay $D_s(i,t)$ and network delay $D_n(i,t)$, namely,

$$D(i,t) = D_s(i,t) + D_n(i,t).$$

Transcoding delay is determined by both the input bit-rate and the target bit-rate [8], while network delay is mainly determined by the network condition between a viewer and a datacenter.

Then we can derive the transcoding delay and the network delay for viewer i at time slot t as follow:

$$D_s(i,t) = \widetilde{D}_s(P(F^t(i),t), B_v(i,t), t),$$

$$D_n(i,t) = \widetilde{D}_n(i, Z(i,t), t).$$

In the above equations, $F^t(i)$ returns the index of a gamer who hosts the channel watched by viewer i at time slot t. And $\widetilde{D}_s(P(F^t(i),t), B_v(i,t), t)$ is the average video transcoding delay for an input version with a bit-rate $P(F^t(i),t)$ and a target version with bit-rate $B_v(i,t)$. $\widetilde{D}_n(i, Z(i,t), t)$ is the network delay between a viewer i and a datacenter $Z(i,t)$ at time slot t.

D. QoE Model

If delay constraints can be satisfied, viewer QoE is mainly determined by the received bit-rate and game genre. Similar to [31] [32], we define the QoE function of viewer i who receives a video stream with a bit-rate $B_v(i,t)$ as $\Psi(G(i), B_v(i,t))$, which is a non-decreasing concave function of received bit-rate $B_v(i,t)$. Such definition can ensure that the marginal benefit brought by increasing the video bit-rate will be diminished when the video bit-rate is high. Thus, in each time slot t, the QoE of viewer i can be defined as below:

$$Q(i,t) = \Psi(G(i), B_v(i,t)).$$

Then, the overall utility of all viewers in the system at time slot t can be represented as:

$$Q(t) = \sum_{i \in V^t} Q(i,t),$$

where V^t stands for the viewer set at time slot t.

E. Problem Formulation

As a CLGVS service provider, it is critical to minimize the operational cost and increase the viewer QoE in the meanwhile. Therefore, our objective in this paper is to design a cost-effective live video transcoding and delivery algorithm. To this purpose, we formulate the problem into the following stochastic optimization problem:

$$\mathbf{P1}.\min \quad \lim_{T\to+\infty} \frac{1}{T}\sum_{t=1}^{T}(C_o(t) - \alpha \cdot Q(t))$$

$$\text{s.t.} \quad \sum_{n=1}^{N} I^t(i,n) = 1, \forall i \tag{a}$$

$$R(j,t) \subseteq \{1,2,\cdots,M\}, \forall j \tag{b}$$

$$B_v(i,t) \in \{B_s(m), m \in R(F^t(i),t)\}, \forall i \tag{c}$$

$$Z(i,t) \in \{1,2,\cdots,N\}, \forall i \tag{d}$$

$$P(j,t) \geq \max\{B_s(m), m \in R(j,t)\}, \forall j \tag{e}$$

$$B_v(i,t) \geq E(G(i)), \forall i \tag{f}$$

$$\lim_{T\to+\infty} \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N^t}\sum_{i\in V^t} D(i,t) \leq \epsilon. \tag{g}$$

In the above problem formulation, the objective function contains two parts. One is the long-term time-average service cost incurred by bandwidth consumption and transcoding service, which should be minimized as much as possible. The other is the total QoE of all viewers, which should be maximized. Obviously, these two components conflict with each other. To optimize two conflicting objective components, we combine them into one weighted-sum objective function. The parameter α is used to tune the trade-off between two components. The values of three decision variables $R(j,t)$, $B_v(i,t)$, $Z(i,t)$ are limited by their corresponding constraints.

Constraint (a) ensures that each viewer receives one stream from only one datacenter. Constraint (b) guarantees that the original stream of gamer j can only be transcoded to M different bit-rates provided by the transcoding cloud. Constraint (c) ensures viewer i can only choose the bit-rate that the video stream has been transcoded to. Constraint (d) makes sure that viewer i can only choose one datacenter from $\{S_1, S_2, \cdots, S_N\}$. Constraint (e) avoids useless transcoding tasks (e.g. transcoding from a lower-bit-rate version to a higher-bit-rate version). And constraint (f) guarantees that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

each viewer receives a bit-rate higher than the basic bit-rate required by the game that the viewer is watching. The last constraint (g) ensures that the average service delay is no greater than a pre-defined threshold. V^t stands for the viewer set at the beginning of time slot t, and N^t is the size of set V^t, which is also the number of online viewers at time slot t. ϵ is the max-tolerable delay experienced by a viewer. When the delay constraint is satisfied, a further reduction on delay brings marginal benefits. In this case, the service provider should improve video quality to increase viewer QoE and reduce operational cost at the same time.

IV. Design of Cost-Effective Online Algorithm

To solve the constrained optimization problem in **P1**, we leverage the Lyapunov optimization framework [11] to design online strategies for transcoding decisions, bit-rate assignment and datacenter selection. A major benefit of the Lyapunov optimization is that it doesn't require any priori knowledge about gamer behaviors, viewer behaviors and network conditions. By taking actions to greedily minimize the drift-plus-penalty in each time slot, it can provide performance with explicit bounds. Other approaches (e.g. Markov decision process [33], non-linear programming [34]) can also be used to solve this problem, but these approaches require some priori knowledge about gamer behaviors, viewer behaviors and network conditions.

In the Lyapunov optimization framework, the original stochastic optimization problem can be transformed into an optimization problem of minimizing the Lyapunov drift-plus-penalty. By using the Lyapunov optimization, the time average delay constraint in Problem **P1** can be transformed into a queue stability constraint [11].

A virtual queue $\mathbf{\Theta}$ is introduced to transform the time average delay constraint into a queue stability constraint. $\mathbf{\Theta}$ represents the virtual delay queue of online viewers and $\mathbf{\Theta}(t)$ denotes the virtual queue backlogs at time slot t. The update of virtual queue $\mathbf{\Theta}(t)$ is given by:

$$\mathbf{\Theta}(t+1) = \max\{\mathbf{\Theta}(t) + \frac{1}{N^t}\sum_{i\in V^t} D(i,t) - \epsilon, 0\}.$$

In Lemma IV.1, we prove that the delay constraint (g) of **P1** can be ensured when the virtual queue $\mathbf{\Theta}$ is stable.

Lemma IV.1 *If the virtual queue $\mathbf{\Theta}$ is stable, then the time average delay constraint (g) of Problem **P1** can be satisfied. That is:*

$$\lim_{T\to+\infty}\frac{\mathbf{\Theta}(T+1)}{T} = 0 \Rightarrow \lim_{T\to+\infty}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N^t}\sum_{i\in V^t} D(i,t) \le \epsilon.$$

Proof: Please see Appendix A in our technical report [40] for the proof details. ■

We define the Lyapunov function as $L(t) = \frac{1}{2}\mathbf{\Theta}^2(t)$. And we use the Lyapunov drift $\Delta(\mathbf{\Theta}(t)) = L(t+1) - L(t)$ to represent the expected change in the Lyapunov function over time. According to the Lyapunov optimization framework, we can then obtain the drift-plus-penalty by adding the objective function in **P1** to the drift, namely,

$$\Delta(\mathbf{\Theta}(t)) + V\mathbb{E}\{C_o(t) - \alpha\cdot Q(t)|\mathbf{\Theta}(t)\},$$

where V is a tunable parameter that affects the performance of the online algorithm. Thus, the solution of the original stochastic optimization problem can be approximately obtained by solving the problem of minimizing the drift-plus-penalty $\Delta(\mathbf{\Theta}(t)) + V\mathbb{E}\{C_o(t) - \alpha\cdot Q(t)|\mathbf{\Theta}(t)\}$ in each time slot. That means, **P1** can be transformed into the following optimization problem:

$$\begin{aligned}\mathbf{P2}.\min\quad & \Delta(\mathbf{\Theta}(t)) + V\mathbb{E}\{C_o(t) - \alpha\cdot Q(t)|\mathbf{\Theta}(t)\}\\ \text{s.t.}\quad & (a)(b)(c)(d)(e)(f).\end{aligned}$$

In order to solve Problem **P2**, we first derive the upper bound of the drift-plus-penalty, denoted by Γ. We give the upper bound Γ in Lemma IV.2.

Lemma IV.2 *For any feasible solution set $R(j,t)$, $B_v(i,t)$, $Z(i,t)$, $I^t(i,n)$, the drift-plus-penalty is upper bounded by Γ, namely*

$$\Delta(\mathbf{\Theta}(t)) + V\mathbb{E}\{C_o(t) - \alpha\cdot Q(t)|\mathbf{\Theta}(t)\} \le \Gamma,$$

where the upper bound $\Gamma = \frac{1}{2}(D_{max}^2 + \epsilon^2) + \mathbb{E}\{\frac{1}{N^t}\mathbf{\Theta}(t)\sum_{i\in V^t} D(i,t)|\mathbf{\Theta}(t)\} + V\mathbb{E}\{C_o(t) - \alpha\cdot Q(t)|\mathbf{\Theta}(t)\}$. *Here we suppose that $D(i,t)$ is a non-decreasing convex function, and upper-bounded by D_{max}.*

Proof: Please see Appendix B in our technical report [40] for the proof details. ■

Instead of minimizing the drift-plus-penalty directly, our strategy actually seeks to minimize the upper bound Γ. By eliminating the constant part and using the rest as the objective function, we transform Problem **P2** into Problem **P3**:

$$\begin{aligned}\mathbf{P3}.\min\quad & \sum_{i\in V^t}\Big(\frac{\mathbf{\Theta}(t)}{N^t}D(i,t) + V\cdot C_b(i,t) - V\alpha\cdot Q(i,t)\Big)\\ & + \sum_{j\in H^t} V\cdot C_s(j,t)\\ \text{s.t.}\quad & (a)(b)(c)(d)(e)(f).\end{aligned}$$

Through a series of conversions, we transform the original optimization problem **P1** into a simple optimization problem **P3**. In order to solve the problem **P3**, we design a one-slot algorithm (as described in **Algorithm 1**), which can obtain the optimal solutions for **P3** and reduce computation complexity significantly.

In **Algorithm 1**, we need to make decisions for each online gamer and viewer at the beginning of each time slot. By leveraging the Lyapunov optimization framework, transcoding decisions, bit-rate assignment and datacenter selection can be derived by solving the one-slot Problem **P3** at each time

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

Algorithm 1 One-Slot Algorithm

Input:

The value of K, M, N, V, α, τ
Number of online viewers N^t
Upload bit-rate of gamer j, $P(j,t)$
Bit-rate mapping function $B_s(m)$
Viewer-category mapping function $G(i)$
Gamer-viewer mapping function $F^t(i)$
Basic bit-rate requirement $E(k)$
Transcoding delay $\widetilde{D}_s(\cdot)$ and network delay $\widetilde{D}_n(\cdot)$
Transcoding cost $\widetilde{C}_s(\cdot)$
QoE function $\Psi(\cdot)$
Delay tolerance ϵ
Unit bandwidth price of datacenter n $U(n,t)$
Queue backlog status $\mathbf{\Theta}(t)$

Output:

Transcoding decisions $R(j,t), \forall j \in H^t$
Bit-rate assignment $B_v(i,t), \forall i \in V^t$
Datacenter selection $Z(i,t), \forall i \in V^t$
Viewer-datacenter indicator $I^t(i,n), \forall i \in V^t$

1: Initialization step: initialize $R(j,t)$, $B_v(i,t)$, $Z(i,t)$, $I^t(i,n)$;
2: **for** all online gamers $j \in H^t$ **do**
3: **for** all online viewers i watching j's channel **do**
4: find optimal solution $(R(j,t), B_v(i,t), Z(i,t), I^t(i,n))$ for **P3**;
5: **end for**
6: **end for**

slot. We can leverage other techniques, such as dynamic programming, to help derive the optimal solutions of Problem **P3**. The details of our online algorithm, called *OCTAD (Online Cloud Transcoding And Distribution)*, are given in **Algorithm 2**.

In Theorem IV.3 we prove that our online algorithm can approach the optimal solution of the original optimization problem within infinitely small distance. And the gap between our online algorithm and the optimal solution is tunable through the parameter V.

Theorem IV.3 *The time-average weighted sum of operational cost and viewer QoE incurred by online algorithm derived via solving Problem **P2** is bounded by $\hat{O} + \frac{\Lambda}{V}$, that is,*

$$\lim_{T\to+\infty} \frac{1}{T}\sum_{t=1}^{T}(C_o(t) - \alpha \cdot Q(t)) \leq \hat{O} + \frac{\Lambda}{V}$$

*where $\hat{O}$ denotes the optimal value of the objective function in Problem **P1**, and $\Lambda = \frac{1}{2}(D_{max}^2 + \epsilon^2)$.*

Proof: Please see Appendix C in our technical report [40] for the proof details. ∎

The tunable parameter V determines the approximation extent of our algorithm to the optimality. The parameter V also determines the tradeoff between the objective function and delay constraints in our problem. With a larger value of V, the performance of our algorithm will be close to the optimality.

Algorithm 2 OCTAD Algorithm

Input:

The value of K, M, N, V, α, τ
Number of online viewers N^t
Upload bit-rate of gamer j, $P(j,t)$
Bit-rate mapping function $B_s(m)$
Viewer-category mapping function $G(i)$
Gamer-viewer mapping function $F^t(i)$
Basic bit-rate requirement $E(k)$
Transcoding delay $\widetilde{D}_s(\cdot)$ and network delay $\widetilde{D}_n(\cdot)$
Transcoding cost $\widetilde{C}_s(\cdot)$
QoE function $\Psi(\cdot)$
Delay tolerance ϵ
Unit bandwidth price of datacenter n $U(n,t)$

Output:

Transcoding decisions, bit-rate assignment, datacenter selection and viewer-datacenter indicators $R(j,t)$, $B_v(i,t)$, $Z(i,t)$ and $I^t(i,n)$ for all online gamers and viewers at each time slot.

1: Initialization step: Let $t = 1$, and set $\mathbf{\Theta}(1) = 0$
2: **while** CLGVS service is operating **do**
3: At the beginning of each time slot t, get the information about queue backlog $\mathbf{\Theta}(t)$ and the real-time bandwidth price $U(n,t)$ of each datacenter n;
4: Gather the information of all online gamers H^t and the uploading bit-rates information of each gamer $P(j,t)$;
5: Get the information of all online viewers V^t and the quantity of online viewers N^t;
6: Update the network delay between each viewer and datacenter $\widetilde{D}_n(\cdot)$;
7: Calculate transcoding decisions, bit-rate assignment, datacenter selection and viewer-datacenter indicators: ($R(j,t), \forall j$; $B_v(i,t), \forall i$; $Z(i,t), \forall i$; $I^t(i,n), \forall i$) by leveraging **Algorithm 1**;
8: Update virtual queues $\mathbf{\Theta}(t)$ according to update operation $\mathbf{\Theta}(t+1) = \max\{\mathbf{\Theta}(t) + \frac{1}{N^t}\sum_{i\in V^t} D(i,t) - \epsilon, 0\}$;
9: Update $t \leftarrow t + 1$;
10: **end while**

However, it is at the cost of a larger virtual queue length, which implies that viewers will experience a larger service delay.

V. Performance Evaluation

In this section, we develop a trace-driven simulator to evaluate the performance of our proposed algorithm. Firstly, we will describe the simulation settings. Next, we compare our proposed algorithm with four other strategies.

A. Simulation Setup

In our simulations, we consider the scenario where a CLGVS service provider deploys its platform over the cloud. We assume that the service provider deploys its streaming servers on five geo-distributed datacenters and leverages cloud transcoding services (e.g. Zencoder[9]) to support live

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

transcoding. To make our simulations more realistic, we adopt the real bandwidth pricing model obtained from the Amazon's website [35]. According to the Amazon's pricing model, bandwidth price is region-dependent and varies over time. In our simulations, we assume that the bandwidth price of each datacenter takes values in the range of $[0.05, 0.07]$ (in units of dollars per gigabyte) and varies over time. We also assume that the bandwidth price of each datacenter is independent and identically distributed (i.i.d.). To support live transcoding service, a CLGVS service provider needs to purchase the transcoding service from a third-party transcoding SaaS provider (e.g. Zencoder [9]). We set the transcoding cost based on the Zencoder's pricing model [36], in which the transcoding cost is determined by three factors: input bit-rate, target bit-rate and video length. In our simulations, we set the transcoding cost in the range of $[0.15, 0.3]$ (in units of dollars per minute).

For simplicity, we assume that gamers broadcast game videos in five different game genres. The service delay tolerance is a tunable parameter of a CLGVS platform. In our experiments, we set the pre-defined threshold of service delay to be 400 milliseconds. Considering that different types of games have various requirements on the video bit-rates to ensure a basic game experience, we assume that the basic bit-rate of each game genre follows a uniform distribution in $[3, 4]$, $[2, 3]$, $[1.5, 2]$, $[1, 1.5]$, $[0.5, 1]$ Mbits/s. According to the upload video bit-rate recommended by YouTube Live [37], we set the upload bit-rate of each channel in the range of $[2, 5]$ Mbits/s. Based on the encoding schemes recommended by Zencoder [38], we suppose a video stream can be transcoded to multiple bit-rates in the range of $[1, 3]$ Mbits/s.

We set the transcoding delay of each channel based on the measurement work in [8]. In our paper, we consider service delay as the sum of network delay and transcoding delay, we set the transcoding delay in the range of $[100, 600]$ milliseconds. Considering that the playout delay is usually constant and will not be affected by the strategy we made, we set the playout delay to be 15 milliseconds according to [39]. We use the real dataset on network delay among Internet nodes obtained from the MIT-King Dataset [41]. In the simulations, the bandwidth of each viewer is configured to be higher than the basic bit-rate of the game that the viewer watches, which means the bandwidth conditions wouldn't affect the basic experience. And we also suppose that the upload bandwidth of a gamer is always higher than the bit-rate of the uploaded game video. In our experiments, we use the dataset in [42] to simulate the behavior of existing gamers and viewers, and the arrival process of new gamers and viewers.

A viewer's QoE is mainly affected by the received video bit-rate, when watching live game videos. Similar to the QoE model in [43], we define the QoE function $\Psi(\cdot)$ as a logarithmic function of the received video bit-rate and the game genre, namely,

$$\Psi(G(i), B_v(i,t)) = \ln(1 + \frac{B_v(i,t)}{E(G(i))}),$$

where $E(G(i))$ is the basic bit-rate required by a viewer i, and $B_v(i,t)$ is the actual bit-rate received by viewer i. In the above definition, the QoE received by a viewer increases concavely with the increase of the ratio between the actual video bit-rate and the basic bit-rate. Such a definition is based on the intuition that, when the received bit-rate is higher than the basic required bit-rate, a further bit-rate increase brings marginal benefits.

Our proposed online algorithm can help a CLGVS service provider make transcoding decisions for each gamer, and can also help on bit-rate adaptation and datacenter selection for each online viewer. In our simulations, we suppose that each time slot lasts for 10 minutes. Since our algorithm makes decisions at the beginning of each time slot, for gamers who begin to broadcast live game in the middle of a time slot, we just transcode the original video to the basic bit-rate. As for the newly-arrived viewers who arrive in the middle of a time slot, we use the *BBS* (which will be described below) strategy to handle their requests timely and dispatch the requests based on the load-balancing principle.

To verify the effectiveness of our algorithm, we compare our algorithm with four other alternatives, each of which consists of a transcoding decision strategy and a bit-rate assignment strategy. For four other alternatives, we suppose that the platform always dispatches user requests based on the load-balancing principle, which means the system is prone to direct user requests to the datacenter with low workload.

In our simulations, we consider two transcoding decision algorithms mentioned in [42], called *Threshold* and *ALL*. Since both strategies have not made decisions on transcoding parameters, the streams of selected channels are transcoded to all the possible bit-rates lower than the original bit-rate in our simulation:

1) *Threshold*, in which a service provider makes transcoding decisions at the beginning of each time slot. Firstly, a service provider filters the gamers with the number of viewers higher than a predefined threshold, and then applies transcoding service to the channels which are hosted by selected gamers. In our simulations, we set the threshold to be 50.
2) *ALL*, in which a service provider applies live transcoding service to all online channels. In this case, a service provider tends to reduce the total bandwidth cost and improve viewer QoE by providing transcoding service to all online channels.

Two typical bit-rate selection strategies mentioned in [44] are used for comparison in our experiments:

1) *Basic Bit-rate Selection (BBS)*, in which viewers are always assigned with the lowest bit-rate that is equal to or just above the basic bit-rate required by the game genre. Under this strategy, a service provider tends to ensure the basic requirement for each viewer, and minimize the operational cost as much as possible.
2) *Advanced Bit-rate Selection (ABS)*, in which a service provider tends to provide better experience by providing a higher bit-rate to each viewer. In our simulations, a service provider assigns a bit-rate that is one level higher than the basic bit-rate for each viewer. Therefore, the service provider can offer a better experience to viewers and probably attract more viewers. However, it may

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

cause a higher operational cost.

Therefore, we have five strategies in comparison, namely, (1) *Threshold + BBS* (2) *Threshold + ABS* (3) *ALL + BBS* (4) *ALL + ABS* (5) our proposed *OCTAD* algorithm.

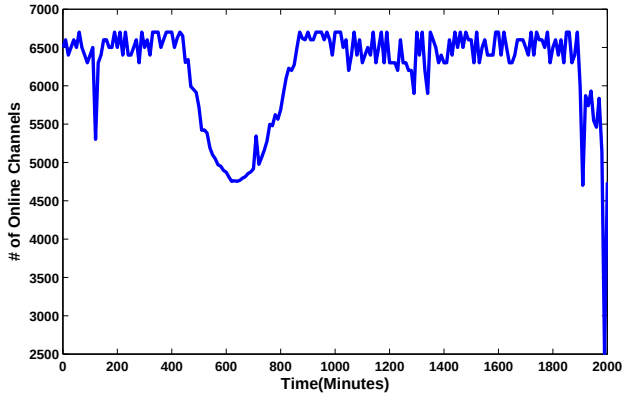


Fig. 2. Number of Online Channels

In our simulations, we use the public dataset [42] to simulate the behaviors of gamers and viewers in each time slot. Fig. 2 depicts the variation of the total number of online channels over $2,000$ minutes. We can observe that the number of online channels is higher than $5,000$ in most of the time. Specifically, the peak value is even higher than $6,500$ channels.

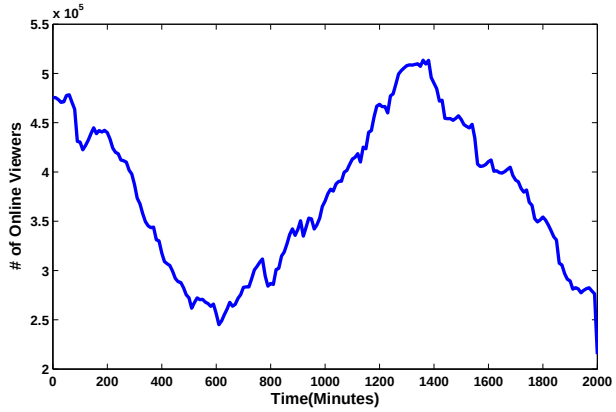


Fig. 3. Number of Online Viewers

Fig. 3 shows the evolution of the total number of online viewers over $2,000$ minutes. From this figure, we can find that in the peak time period, there are about $510,000$ online viewers in the system. However, the total number of online viewers varies dramatically over time, and the peak-to-valley gap is about $250,000$.

B. Comparison of Operational Cost

Fig. 4 depicts the total operational cost incurred by six strategies in the simulation period. From this figure, we can find that *ALL + ABS* incurs the highest operational cost, and our *OCTAD* algorithm achieves the lowest operational cost. The operational cost incurred by our *OCTAD* algorithm is close to

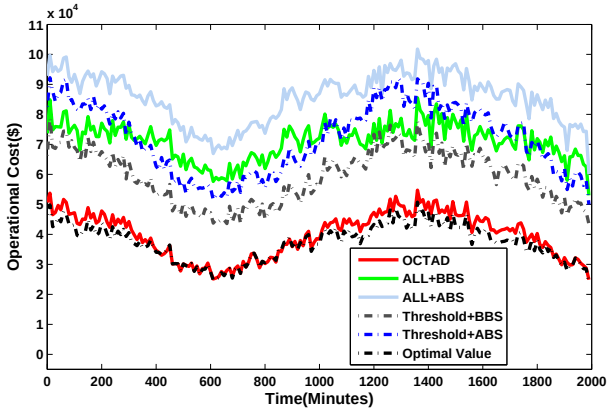


Fig. 4. Operational Cost

the optimal value with very small distance. Specifically, our algorithm can reduce operational cost by 50% compared with *ALL + ABS*, by 40% compared with *Threshold + ABS*, by 35% compared with *ALL + BBS* and by 25% compared with *Threshold + BBS*.

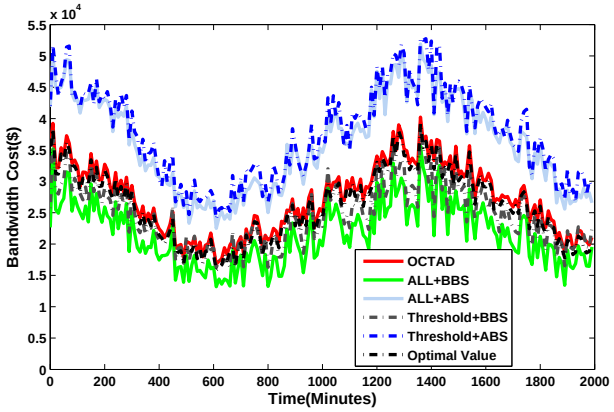


Fig. 5. Bandwidth Cost

The CLGVS service provider not only streams live game videos to viewers, but also needs to transcode gamers' original streams to multiple versions. Therefore, the operational cost consists of two parts, bandwidth cost and transcoding cost. The comparison of bandwidth cost is shown in Fig. 5. From this figure, we can find that the bandwidth cost of *ALL + BBS* is the lowest among all strategies. According to our definition, bandwidth cost is determined by the bandwidth price and bandwidth consumption in each datacenter. Since *ALL + BBS* assigns viewers with the basic bit-rate or a bit-rate just above the basic bit-rate, it can achieve the lowest bandwidth cost. The bandwidth cost incurred by *Threshold + BBS* is a little bit higher than that of *ALL + BBS*. It is because that, although *Threshold + BBS* and *ALL + BBS* apply the same bit-rate selection algorithm, channels with viewers lower than predefined threshold are not transcoded to the basic bit-rate, and these viewers have no choice but select the original bit-rate. As for *ALL + ABS* and *Threshold + ABS*, both of them are the highest among all strategies, because they use the same

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

bit-rate selection algorithm and always select a higher bit-rate for viewers. By taking various factors into consideration when making decisions, our *OCTAD* algorithm achieves a rather low bandwidth cost, which is close to the lowest value.

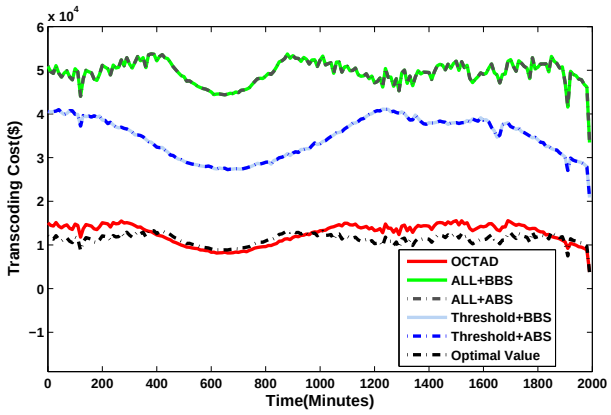


Fig. 6. Transcoding Cost

Fig. 6 depicts the transcoding cost under five different strategies. In this figure, we can find that *ALL + BBS* and *ALL + ABS* incur the same transcoding cost and also the highest among all strategies. Because these two combined algorithms both apply the live transcoding service to all online channels without taking other factors (e.g. channel popularity, game genre, etc.) into account. *Threshold + BBS* and *Threshold + ABS* just provide the live transcoding service to some popular channels. Therefore, they incur relatively low transcoding cost. The transcoding cost incurred by our *OCTAD* algorithm is close to that of the optimal solution. More specifically, our *OCTAD* algorithm cuts down more than 60% of transcoding cost compared with both *ALL + BBS* and *ALL + ABS*. In addition, *OCTAD* algorithm reduces over 50% of transcoding cost compared with *Threshold + BBS* and *Threshold + ABS*. These results prove that our proposed *OCTAD* algorithm can reduce transcoding cost significantly for the service provider.

C. Comparison of Viewer QoE

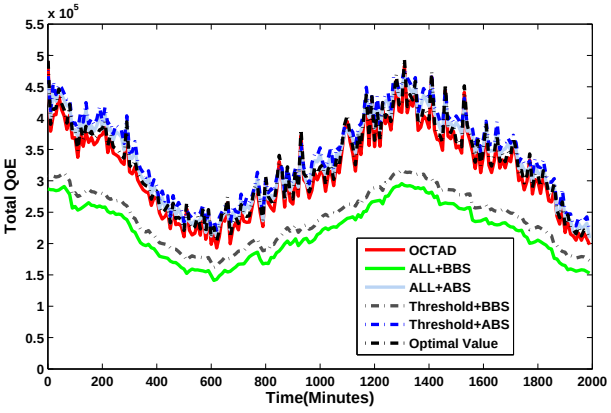


Fig. 7. Viewer QoE

In Fig. 7, we compare the viewer QoE under various strategies in each time slot. *ALL + ABS* and *Threshold + ABS* can provide the highest viewer QoE since these two strategies always select a higher bit-rate, which brings much better viewer QoE. While *ALL + BBS* and *Threshold + BBS* are both prone to choose the basic bit-rate, which can only ensure the basic experience. Thus, viewers under these two strategies gain lower QoE. Moreover, *ALL + BBS* provides the lowest bit-rate. Although, *Threshold + BBS* adopts the same bit-rate assignment algorithm as *ALL + BBS*. Under the transcoding strategy *Threshold*, the service provider just applies live transcoding to top channels. In this case, viewers who are watching less-popular channels can just receive raw streams. That is why *Threshold + BBS* achieves a little bit higher QoE compared with *ALL + BBS*. Finally, our *OCTAD* algorithm can achieve almost the same QoE as *ALL + ABS* and *Threshold + ABS*, which is much higher than QoE achieved by *ALL + BBS* and *Threshold + BBS*.

D. Comparison of Delays

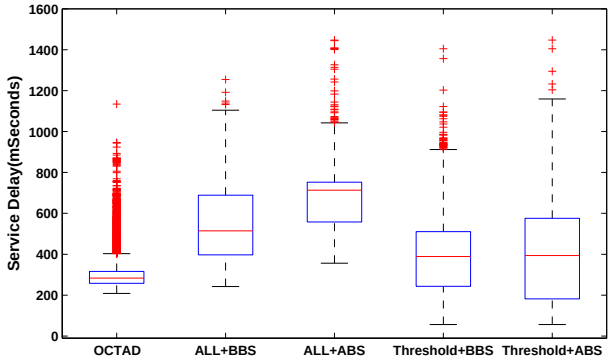


Fig. 8. Service Delay

Fig. 8 shows the comparison of service delay under different methods. From the figure, we can observe that our proposed algorithm can keep service delay in a relatively low level. For 75% of viewers, their service latency is lower than 300 milliseconds, which is tolerable for viewers who are watching live videos. Most of the viewers experience almost the same service delay, which proves that *OCTAD* algorithm can guarantee fairness among viewers. When the delay constraint is satisfied, a further reduction of delay brings marginal benefits.

In our formulations, we define the service delay as the sum of network delay and transcoding delay. Fig. 9 depicts the comparison of transcoding delay under five strategies. Among all strategies, *ALL + ABS* results in the highest transcoding delay. Because *ALL + ABS* applies the live transcoding service to all online channels and transcodes the original streams to higher video bit-rates. According to the measurement work in [8], a higher target bit-rate will cause a higher transcoding delay. Furthermore, *Threshold + BBS* and *Threshold + ABS* can achieve the lowest transcoding delay. It is because these algorithms just provide transcoding service to some top chan-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

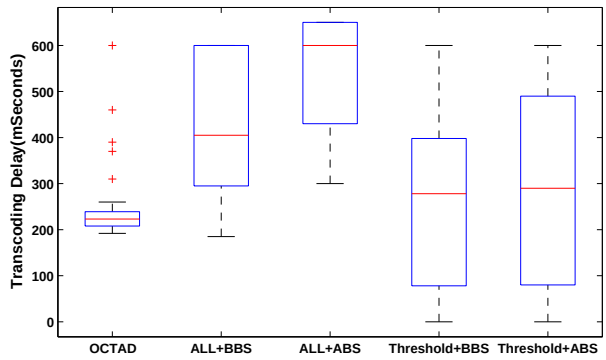


Fig. 9. Transcoding Delay

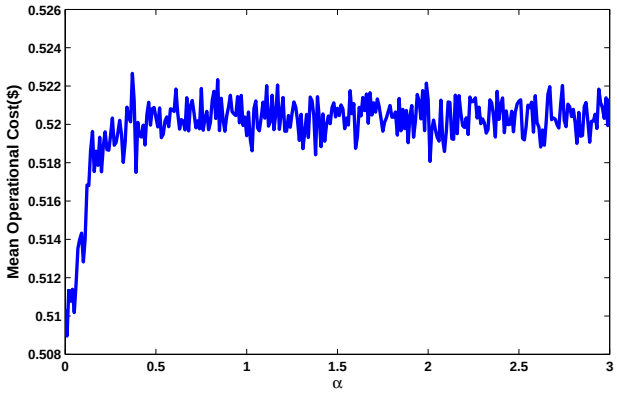


Fig. 11. Mean Operational Cost Under Various α

nels. However, by making transcoding decisions dynamically, the transcoding delay under *OCTAD* algorithm is tolerable.

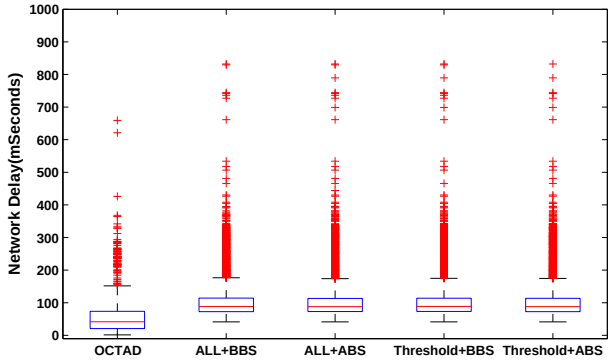


Fig. 10. Network Delay

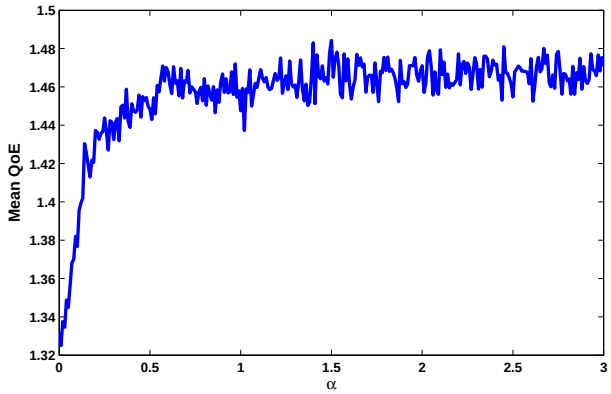


Fig. 12. Mean Viewer QoE Under Various α

Fig. 10 displays the network latency under five different strategies. For each viewer, the network delay is determined by the network condition between a datacenter and a viewer. Since the other four alternatives dispatch viewer requests to a datacenter with the lowest workload without considering the network conditions, these strategies incur much higher network latency. Our proposed algorithm takes network conditions into account when making datacenter selection. It is the reason why our algorithm can achieve a much lower network delay compared with other alternatives.

E. Impacts of Tunable Parameters

In our proposed algorithm, there are two tunable parameters α and V. To investigate the impacts of these two parameters, we conduct some further experiments. When making decisions, the value of α affects the tradeoff between operational cost and viewer QoE. Firstly, we conduct a series of experiments with various values of α to examine the influence of α. From Fig. 11, we can observe that, when the value of α increases from 0 to 3, the mean operational cost increases from $0.508\$$ to $0.521\$$. Fig. 12 shows that mean QoE increases with the increase of α. When the value of α increases from 0 to 3, the mean QoE increases from 1.32 to 1.47. Intuitively, a higher value of α allocates more weight to viewer QoE when *OCTAD* algorithm makes decisions, which will improve viewer QoE. However, the side effect of a higher QoE is a higher operational cost due to the increasing resource consumption. When the value of α is higher than 0.6, the impact of α tends to be stable. This means that even when we keep increasing α, the influence of α is negligible. Anyway, a higher value of α brings better viewer experience. It indicates that we should choose the value of α skillfully, as a higher α incurs a higher operational cost.

Another tunable parameter of *OCTAD* algorithm is V. According to the Lyapunov optimization theory, V is a tunable parameter which affects both virtual queue backlog and the Lyapunov penalty. In this paper, the Lyapunov penalty consists of operational cost and viewer QoE. Fig. 13 depicts the influence of the parameter V on the mean operational cost. In Fig. 13, when V increases from 0 to 2000, the mean operational cost decreases from $0.44\$$ to $0.32\$$. In addition, viewer QoE is another important system performance indicator. In Fig. 14, when V increases from 0 to 2000, the mean viewer QoE keeps rising from 1.08 to 1.2. But when we further increase the value of V, the mean QoE will oscillate in a small range. This means that the impact of V tends to be stable. Although with a higher value of V, our algorithm can approach to the optimal value, a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2556584, IEEE Transactions on Circuits and Systems for Video Technology

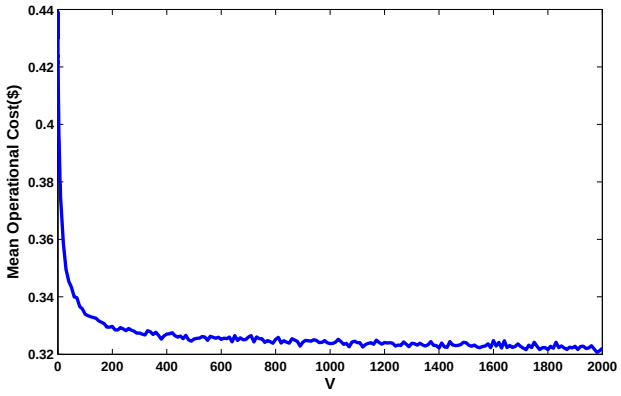


Fig. 13. Mean Operational Cost Under Various V

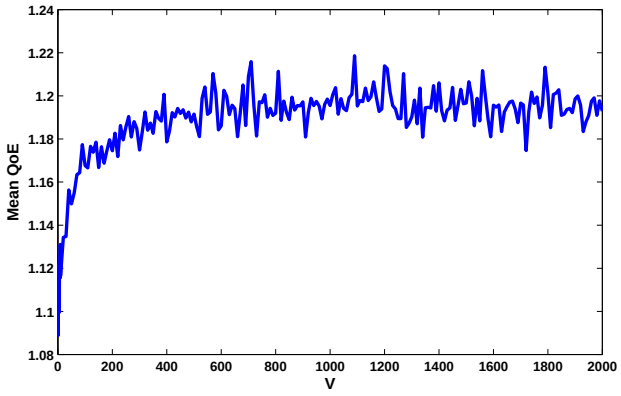


Fig. 14. Mean Viewer QoE Under Various V

large value of V will increase the service latency significantly.

VI. Conclusion and Future Work

In this paper, we investigated the problem of providing cost-effective live transcoding and game video delivery service to gamers and viewers from the perspective of a CLGVS service provider. Because the objectives of reducing operational cost and improving viewer QoE are conflicting with each other, it is hard to make optimal decisions. We formulated the problem into a constrained stochastic optimization problem, whose objective is to minimize operational cost while still ensure good-enough service quality. By leveraging the Lyapunov optimization framework, we derived an adaptive online algorithm called OCTAD, which can help service providers make live transcoding decisions, bit-rate assignment and datacenter selection dynamically. Through a series of trace-driven simulations, we evaluated the effectiveness of OCTAD. Compared with other algorithms, our OCTAD algorithm can reduce operational cost significantly while still guarantee good-enough viewer experience. In our future work, we plan to implement our algorithm in a real live game video streaming platform, and try to combine live streaming with a cloud gaming platform.

References

[1] (2015) *Twitch.tv*. [Online]. http://www.twitch.tv/.

[2] (2014) *Twitch.tv Official Blog*. [Online]. http://blog.twitch.tv/2014/02/twitch-community-4th-in-peak-us-internet-traffic/.

[3] R. Shea, D. Fu, and J. Liu, "Towards bridging online game playing and live broadcasting: Design and optimization," in *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, ser. NOSSDAV '15. ACM, 2015, pp. 61–66.

[4] C. Zhang and J. Liu, "On crowdsourced interactive live streaming: A twitch.tv-based measurement study," in *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, ser. NOSSDAV '15. ACM, 2015, pp. 55–60.

[5] (2015) *Youtube*. [Online]. https://www.youtube.com/.

[6] (2015) *Hulu*. [Online]. http://www.hulu.com/.

[7] (2015) *Douyu.tv*. [Online]. http://www.douyu.tv/.

[8] R. Aparicio-Pardo, K. Pires, A. Blanc, and G. Simon, "Transcoding live adaptive video streams at a massive scale in the cloud," in *Proceedings of the 6th ACM Multimedia Systems Conference*, ser. MMSys '15. ACM, 2015, pp. 49–60.

[9] (2015) *Zencoder*. [Online]. https://zencoder.com/.

[10] (2015) *Encoding.com*. [Online]. https://www.encoding.com/.

[11] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[12] K. Pires and G. Simon, "Youtube live and twitch: A tour of user-generated live streaming systems," in *Proceedings of the 6th ACM Multimedia Systems Conference*, ser. MMSys '15. ACM, 2015, pp. 225–230.

[13] T. Smith, M. Obrist, and P. Wright, "Live-streaming changes the (video) game," in *Proceedings of the 11th European Conference on Interactive TV and Video*, ser. EuroITV '13. ACM, 2013, pp. 131–138.

[14] (2105) *OnLive*. [Online]. http://onlive.com/.

[15] W. A. Hamilton, O. Garretson, and A. Kerne, "Streaming on twitch: Fostering participatory communities of play within live mixed media," in *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '14. ACM, 2014, pp. 1315–1324.

[16] Z. Wang, L. Sun, C. Wu, W. Zhu, and S. Yang, "Joint online transcoding and geo-distributed delivery for dynamic adaptive streaming," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 91–99.

[17] J.-C. Wu, P. Huang, J. Yao, and H. Chen, "A collaborative transcoding strategy for live broadcasting over peer-to-peer iptv networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 2, pp. 220–224, 2011.

[18] R. Cheng, W. Wu, Y. Lou, and Y. Chen, "A cloud-based transcoding framework for real-time mobile video conferencing system," in *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2014 2nd IEEE International Conference on*. IEEE, 2014, pp. 236–245.

[19] Y. Jin, Y. Wen, and C. Westphal, "Optimal transcoding and caching for adaptive streaming in media cloud: an analytical approach," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 25, no. 12, pp. 1914–1925, 2015.

[20] G. Gao, W. Zhang, Y. Wen, Z. Wang, and W. Zhu, "Towards cost-efficient video transcoding in media cloud: Insights learned from user viewing patterns," *Multimedia, IEEE Transactions on*, vol. 17, no. 8, pp. 1286–1296, 2015.

[21] V. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1620–1628.

[22] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, "A measurement study of a large-scale p2p iptv system," *Multimedia, IEEE Transactions on*, vol. 9, no. 8, pp. 1672–1687, 2007.

[23] F. Wang, J. Liu, M. Chen, and H. Wang, "Migration towards cloud-assisted live media streaming," *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[24] J. He, D. Wu, Y. Zeng, X. Hei, and Y. Wen, "Toward optimal deployment of cloud-assisted video distribution services," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 10, pp. 1717–1728, 2013.

[25] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. Lau, "Scaling social media applications into geo-distributed clouds," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 684–692.

[26] Y. Xu, C. Yu, J. Li, and Y. Liu, "Video telephony for end-consumers: Measurement study of google+, ichat, and skype," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. ACM, 2012, pp. 371–384.

[27] (2015) *Panda*. [Online]. https://www.pandastream.com/.

[28] F. Chen, C. Zhang, F. Wang, and J. Liu, "Crowdsourced live streaming over the cloud," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 2524–2532.

[29] K.-T. Chen, Y.-C. Chang, P.-H. Tseng, C.-Y. Huang, and C.-L. Lei, "Measuring the latency of cloud gaming systems," in *Proceedings of the 19th ACM International Conference on Multimedia*, ser. MM '11. ACM, 2011, pp. 1269–1272.

[30] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, ser. IMW '02. ACM, 2002, pp. 5–18.

[31] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "Qoe-driven cache management for http adaptive bit rate streaming over wireless networks," *Multimedia, IEEE Transactions on*, vol. 15, no. 6, pp. 1431–1445, 2013.

[32] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience," *Telecommunication Systems*, vol. 52, no. 2, pp. 587–600, 2011.

[33] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. John Wiley & Sons, Inc., 1994.

[34] M. S. Bazaraa, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. Wiley Publishing, 2013.

[35] (2015) *Amazon EC2 pricing*. [Online], https://aws.amazon.com/ec2/pricing/.

[36] (2015) *Zencoder pricing*. [Online], https://zencoder.com/en/live-transcoding/pricing/.

[37] (2015) *Youtube Help*. [Online]. https://support.google.com/youtube/answer/2853702?hl=en/.

[38] (2015) *Zencoder Hls Guide*. [Online]. https://zencoder.com/en/hls-guide.

[39] C.-Y. Huang, K.-T. Chen, D.-Y. Chen, H.-J. Hsu, and C.-H. Hsu, "Gaminganywhere: The first open source cloud gaming system," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 1s, pp. 10:1–10:25, 2014.

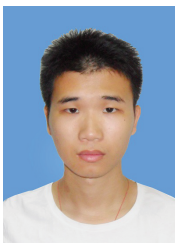
[40] Y. Zheng, D. Wu, Y. Ke, C. Yang, M. Chen, and G. Zhang, "Online Cloud Transcoding and Distribution for Crowdsourced Live Game Video Streaming," Department of Computer Science, Sun Yat-sen University, Tech. Rep., 2016. [Online]. Available: http://netlab.sysu.edu.cn/~dwu/octad-tr.pdf

[41] (2005) *The King dataset*. [Online], http://pdos.csail.mit.edu/p2psim/kingdata/.

[42] K. Pires and G. Simon, "Dash in twitch: Adaptive bitrate streaming in live game streaming platforms," in *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, ser. VideoNext '14. ACM, 2014, pp. 13–18.

[43] J. He, Y. Wen, J. Huang, and D. Wu, "On the cost-qoe tradeoff for cloud-based video streaming uder amazon ec2's pricing models," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, no. 4, pp. 669–680, 2014.

[44] H. Tian, D. Wu, J. He, Y. Xu, and M. Chen, "On achieving cost-effective adaptive cloud gaming in geo-distributed data centers," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 25, no. 12, pp. 2064–2077, 2015.



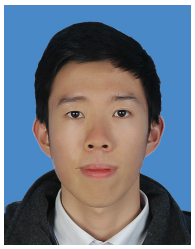
Yuanhuan Zheng received the B.S. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2014. He is currently pursuing the masters degree in the Department of Computer Science, Sun Yat-sen University, Guangzhou, China, under the supervision of Prof. D. Wu.

His research interests include content distribution network, multimedia networking and data center network.

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 Assistant Dean of the School of Data and 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.



Yihao Ke received the B.S. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2014. Currently, he is a graduate student in the Department of Computer Science, Sun Yat-sen University, Guangzhou, China.

His research interests mainly include cloud computing, cloud gaming, cloud storage and GPU virtualization.

Can Yang received his Ph.D., M.S. and B.S. degree in 2002, 1997 and 1994, respectively, and was a Post-Doctoral fellow from 2002 to 2004, all in Huazhong University of Science and Technology. He has been an associate professor and the director of new media lab since 2005, and is currently an advanced professor at the School of Computer Science and Engineering of South China University of Technology. He was a visiting scholar with Electrical and Computer Engineering at the Polytechnic School of Engineering of New York University from 2013 to 2014.

His interest focuses on multimedia computing, peer to peer networking, information theory and coding. He won the first level of the Science and Technology Progressive Award of Canton in 2015 and the second level in 2012, and also the Sci. and Tech. Innovation Award of SARFT of China in 2012. He have published mort than 50 papers and invented 18 Chinese patents. He is a member of IEEE and ACM and a senior member of CCF.

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.

Guoqing Zhang received his Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, where he is currently a principal investigator.

His research interests include information networks and network science, especially topology analysis of the Internet, online social networks and big data, as well as topology-aware optimization and application.