Joint Transcoding Task Assignment and Association Control for Fog-assisted Crowdsourced Live Streaming

PLEASE CITE THE PUBLISHED VERSION

https://doi.org/10.1109/lcomm.2019.2934095

PUBLISHER

Institute of Electrical and Electronics Engineers (IEEE)

VERSION

AM (Accepted Manuscript)

REPOSITORY RECORD

Liu, Xingchi, Mahsa Derakhshani, and Sangarapillai Lambotharan. 2019. “Joint Transcoding Task Assignment and Association Control for Fog-assisted Crowdsourced Live Streaming”. figshare. https://hdl.handle.net/2134/9773639.v1.

Joint Transcoding Task Assignment and Association Control for Fog-assisted Crowdsourced Live Streaming

Xingchi Liu, Mahsa Derakhshani, *Member, IEEE* and Sangarapillai Lambotharan, *Senior Member, IEEE*

***Abstract*—The rapid development of content delivery networks and cloud computing has facilitated crowdsourced live-streaming platforms (CLSP) that enable people to broadcast live videos which can be watched online by a growing number of viewers. However, in order to ensure reliable viewer experience, it is important that the viewers should be provided with multiple standard video versions. To achieve this, we propose a joint fog-assisted transcoding and viewer association technique which can outsource the transcoding load to a fog device pool and determine the fog device with which each viewer will be associated, to watch desired videos. The resulting non-convex integer programming has been solved using a computationally attractive complementary geometric programming (CGP). The performance of the proposed algorithm closely matches that of the globally optimum solution obtained by an exhaustive search. Furthermore, the trace-driven simulations demonstrate that our proposed algorithm is able to provide adaptive bit rate (ABR) services.**

***Index Terms*—Crowdsourced systems, live video transcoding, fog computing, viewer association.**

I. INTRODUCTION

The current crowdsourced live streaming has allowed a growing number of people start to broadcast live videos via the CLSPs such as Twitch.TV, Douyu.TV, and Periscope. Due to heterogeneity of broadcasters' devices, different quality versions of live videos need to be created and uploaded to the CLSP [1]. As a result, there is a strong need to transcode the original live videos into several industrial standard representations and to serve viewers with a set of proper versions of representations, which is referred to as the ABR service.

To provide ABR service, there are massive computational demands due to real-time processing requirements which need to be met. For instance, Twitch TV supported 35610 concurrent broadcasters and over two million viewers during peak time in 2015. Therefore, cloud computing has become a natural solution to perform transcoding because of its powerful computing ability and the 'pay as you go' feature. Previous works have mainly focused on designing a cloud transcoding system (e.g., [2]–[7]). In such a system, the controller is able to decide the number of representations that need to be transcoded for each broadcaster based on parameters such as viewer capacity, playback delay, bandwidth consumption etc.

On the downside, cloud transcoding is not able to provide the ABR service to most of the broadcasters. For instance, in Twitch.TV, only the premium broadcasters have access to the ABR service, and for the rest of the broadcasters, only the original versions are available. The reason behind this is that a general cloud instance can only deal with at most two transcoding tasks simultaneously. Therefore, an enormous cost will be incurred when a large number of original live videos is planned to be transcoded. Moreover, in cloud transcoding systems, the cloud data center can be far from the viewers or the broadcasters. This will cause high latency. In addition, most of CLSPs support the broadcasters and viewers with interactive chat service. Under such a scenario, the latency problem has become even more significant than the traditional live streaming platforms.

Because fog computing [8] is more suitable for real-time processing and low-latency applications, it can be treated as a viable replacement (e.g., [9]–[12]) to address the weakness of the cloud transcoding. In [10], a case study is presented for Twitch.TV. This case demonstrates that with the advance of personal computing devices, a significant fraction of CLSP viewers potentially has appropriate computing resources for stable real-time transcoding. In addition, the viewers have already expressed the willingness to support the broadcasters and the CLSPs in terms of donation and subscription [12]. Thus, the cost by involving them into transcoding can be much lower. Moreover, fog-assisted transcoding can lead to lower latency and avoid the network traffic traversing through the core network since different versions of videos will be created at the network edge.

With fog-assisted transcoding, it is a challenge to dynamically assign transcoding tasks to stable and efficient fog devices since they are not dedicated for transcoding and they could even be offline during transcoding [11]. Therefore, we design a fog-assisted transcoding algorithm which aims to assign the transcoding tasks to proper fog devices and optimize the tradeoff between the QoE of viewers and the cost of transcoding. The technical and novel contributions of the paper can be summarized as

- The transcoding success rates of the fog devices are introduced and a new QoE metric is defined for the fog-assisted transcoding system.
- To the best of our knowledge, this paper is the first work to consider a joint transcoding task assignment and viewer

---

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/R006385/1.

X. Liu, M. Derakhshani, and S. Lambotharan are with the Signal Processing and Networks Research Group, Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough LE11 3TU, U.K. (email:X.Liu3@lboro.ac.uk; M.Derakhshani@lboro.ac.uk; S.Lambotharan@lboro.ac.uk).

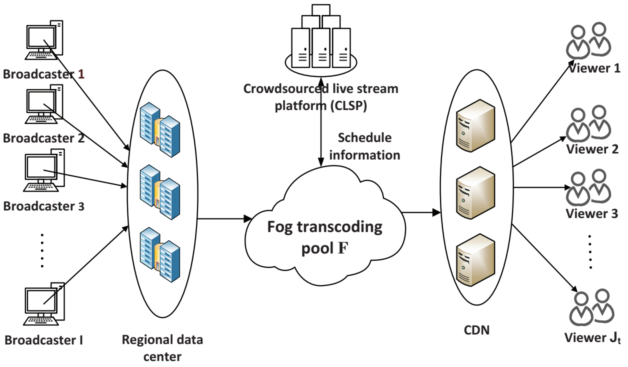

Fig. 1: System model

association problem with fog computing.

- We develop an optimization algorithm using CGP to solve the resultant non-convex integer programming.
- Trace-driven simulations demonstrate that the proposed algorithm outperforms existing benchmark schemes and can dynamically decide the transcoding schedule.

II. SYSTEM MODEL

Assume there is a set of broadcasters, i.e., $\mathcal{I} = \{1, 2, \cdots, I\}$ in the network as illustrated in Fig. 1. We consider a time-slotted system. The bit rate of the original live video of broadcaster $i \in \mathcal{I}$ in time slot t is defined as $B_i(t)$. Moreover, consider there are J_t viewers in time slot t and $\mathbf{V}_{i,j}(t)$ is a binary parameter indicating whether viewer $j \in \mathcal{J}(t) = \{1, 2, \cdots, J_t\}$ chooses to watch broadcaster i's live stream in time slot t or not. Denote $r \in \mathcal{R} = \{1, 2, \cdots, R\}$ as a video representation which is one of R possible standard quality levels of a transcoded video. The bit rate of representation r is defined as b_r.

When an original live video is uploaded by a broadcaster i to the regional data centers, the CLSP scheduler will decide which representation should be transcoded by which fog device based on the viewer requirements. Then, all the original live videos will be transmitted to the selected fog devices through those regional data centers for transcoding. After the transcoding process, all the standard transcoded live videos will be transmitted from the fog devices to the associated viewers. Let us assume $\mathcal{F}$ represents the set of all available fog devices. It is assumed that each fog device $f \in \mathcal{F} = \{1, 2, \cdots, F\}$ is able to transcode one broadcaster's live video into only one standard video version because of the limited computational capability of fog devices.

To describe the transcoding task assignment and viewer association problem, two binary variables are defined as $\mathbf{I}_{i,r,f}(t)$ and $\mathbf{W}_{j,f}(t)$. $\mathbf{I}_{i,r,f}(t)$ takes 1 when the fog device f is selected to transcode the original video from broadcaster i into representation r in time slot t and 0 otherwise. In addition, $\mathbf{W}_{j,f}(t)$ takes 1 if viewer j is associated with fog device f in time slot t to watch the transcoded live video and 0 otherwise.

A. *Cost model*

To incentivize a fog device to participate in transcoding, CLSP will pay a reward to a fog device if the assigned transcoding task is successfully accomplished. The cost for the task of transcoding broadcaster i video into representation r by fog node f is defined as

$$c_{i,r,f}(t) = \mathbf{I}_{i,r,f}(t) \cdot \phi(r) \cdot \alpha_f, \tag{1}$$

where $\phi(r)$ is a non-decreasing concave function of r to model the fact that transcoding the same live video to a higher version of representation consumes more computing resources thus causes higher costs. The reward is paid for the transcoding task only once regardless of how many viewers are watching the same video representation from a broadcaster. Moreover, α_f is defined as the transcoding success rate of device f which reflects the online transcoding stability of f. A higher value of α_f means the probability of being offline during the transcoding is smaller. The total cost related to broadcaster i is defined as

$$c_i(t) = \sum_{r \in \mathcal{R}} \sum_{f \in \mathcal{F}} c_{i,r,f}(t). \tag{2}$$

B. *QoE model*

From the perspective of a viewer, the QoE is determined by two factors. First, the acceptable quality levels of the received live videos of different broadcasters vary in terms of their genres (e.g., card game, pixel art game, first shooter game, etc.) By categorizing the live videos into a set of genres denoted by $\mathcal{G} = \{1, 2, \ldots, K\}$ and defining $g_i(t) \in \mathcal{G}$ as the genre of video from broadcaster i in time slot t, we can define $s_{g_i(t)}$ as the basic bit rate that a broadcaster i is suggested to set for the live video. Second, it is vital to consider the network capacity of each viewer. Let $u_j(t)$ be the highest bit rate that viewer j can receive, which varies due to the viewer network condition. Therefore, the QoE model can be expressed as

$$q_{i,j,r}(t) = \log\left(\frac{b_r}{u_j(t)} + \frac{b_r}{s_{g_i(t)}}\right). \tag{3}$$

In (3), the QoE model is a non-decreasing concave function of two ratios. The first ratio quantifies the effect of the network condition of viewer j. The higher this ratio is, the better QoE can be achieved. However, this ratio should not exceed one, and a constraint is added to the optimization problem; Otherwise, the viewer capacity is smaller than the bit rate of the transcoded representation and this transcoded representation cannot be smoothly played at the viewer end. The second ratio quantifies how better the received video quality is compared with the basic genre rate of broadcaster i considering the fact that same representation from different genres of broadcasters can lead to different quality of experience levels. Next, the QoE for a viewer j watching a transcoded video from broadcaster i can be calculated as

$$Q_{j,i}(t) = \sum_{r \in \mathcal{R}} \sum_{f \in \mathcal{F}} \mathbf{V}_{i,j}(t)\mathbf{I}_{i,r,f}(t)\mathbf{W}_{j,f}(t)\alpha_f q_{i,j,r}(t). \tag{4}$$

C. *Network utility*

There is a tradeoff between the viewer QoE and the cost imposed on CLSP. On one hand, CLSP prefers to incentivize more fog devices to participate in transcoding and provide ABR service to more viewers. On the other hand, better service will incur more cost. To better express this tradeoff, we define a weighted-difference between the QoE and the cost as

$$U_i(t) = \sum_{j\in\mathcal{J}(t)} Q_{j,i}(t) - \lambda \cdot c_i(t), \tag{5}$$

where U_i denotes the network utility in which the parameter λ is used to tune the tradeoff between the two components.

III. FOG TRANSCODING AND VIEWER ASSOCIATION

A. Problem formulation

In this work, we aim to jointly optimize the transcoding task assignment and viewer association by maximizing the total network utility in each time slot over the whole transcoding system. We therefore, formulate an optimization problem as

$$(\mathcal{P}) \quad \max_{\mathbf{I}_{i,r,f}(t),\mathbf{W}_{j,f}(t)} \sum_{i\in\mathcal{I}} U_i(t), \tag{6a}$$

$$\textbf{s.t.} \quad \mathbf{C1}: \ \mathbf{V}_{i,j}(t)\mathbf{I}_{i,r,f}(t)\mathbf{W}_{j,f}(t)b_r \le \min\{u_j(t), B_i(t)\}\ , \tag{6b}$$

$$\mathbf{C2}: \ \sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{I}} \mathbf{I}_{i,r,f}(t) \le 1 \quad \forall f\in\mathcal{F}, \tag{6c}$$

$$\mathbf{C3}: \ \sum_{f\in\mathcal{F}} \mathbf{I}_{i,r,f}(t) \le 1 \quad \forall i\in\mathcal{I}, \forall r\in\mathcal{R}, \tag{6d}$$

$$\mathbf{C4}: \ \sum_{f\in\mathcal{F}} \mathbf{W}_{j,f}(t) = 1 \quad \forall j\in\mathcal{J}(t), \tag{6e}$$

$$\mathbf{C5}: \ \mathbf{I}_{i,r,f}(t) \in \{0,1\} \ \forall r\in\mathcal{R}, \forall i\in\mathcal{I}, \forall f\in\mathcal{F}, \tag{6f}$$

$$\mathbf{C6}: \ \mathbf{W}_{j,f}(t) \in \{0,1\} \ \forall j\in\mathcal{J}(t), \forall f\in\mathcal{F}, \tag{6g}$$

where **C1** ensures the received bit rate is lower than both the original video bit rate and the viewer capacity. **C2** ensures that each fog device can at most transcode one specific original live video into one representation. **C3** guarantees that each transcoding task can only be assigned to one fog device. **C4** makes sure that each viewer only receives one video representation in one time slot. **C5** and **C6** guarantee that variable $\mathbf{I}_{i,r,f}(t)$ and $\mathbf{W}_{j,f}(t)$ are binary.

$\mathcal{P}$ is a non-convex integer programming which is inherently complex to solve. We transform $\mathcal{P}$ by first relaxing the binary variables $\mathbf{I}_{i,r,f}(t)$ and $\mathbf{W}_{j,f}(t)$ to continuous variables and then introducing an auxiliary variable b such that

$$b \le \sum_{i\in\mathcal{I}} U_i(t). \tag{7}$$

Since b is the lower bound of the objective function, maximizing b is equal to maximizing the total network utility. Consequently, with this auxiliary variable and the new constraint in (7), the original problem will be transformed and expressed as

$$(\widetilde{\mathcal{P}}) \quad \max_{\mathbf{I}_{i,r,f}(t),\mathbf{W}_{j,f}(t),b} \quad b, \quad \textbf{s.t.} \quad \mathbf{C1}-\mathbf{C4} \tag{8a}$$

$$\widetilde{\mathbf{C5}}: \mathbf{I}_{i,r,f}(t) \in [0,1] \ \forall r\in\mathcal{R}, \forall i\in\mathcal{I}, \forall f\in\mathcal{F}, \tag{8b}$$

$$\widetilde{\mathbf{C6}}: \mathbf{W}_{j,f}(t) \in [0,1] \ \forall j\in\mathcal{J}(t), \forall f\in\mathcal{F}, \tag{8c}$$

$$\mathbf{C7}: b > 0, \tag{8d}$$

$$\mathbf{C8}: \frac{b + \lambda\sum_{i\in\mathcal{I}}\sum_{r\in\mathcal{R}}\sum_{f\in\mathcal{F}} c_{i,r,f}(t)}{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}(t)}\sum_{r\in\mathcal{R}}\sum_{f\in\mathcal{F}} q_{i,j,r,f}(t)} \le 1, \tag{8e}$$

where $q_{i,j,r,f}(t) = \alpha_f q_{i,j,r}(t)\mathbf{V}_{i,j}(t)\mathbf{I}_{i,r,f}(t)\mathbf{W}_{j,f}(t)$. The problem $\widetilde{\mathcal{P}}$ belongs to the category of CGP since the non-convex constraint of **C8** is in the form of a ratio between two posynomials.

In order to solve this problem, we use successive convex approximation method and monomial approximations [13] to transform the problem into a series of GP problems. Applying monomial approximations, the problem can be expressed as

$$(\widehat{\mathcal{P}}) \quad \max_{\mathbf{I}_{i,r,f}(t),\mathbf{W}_{j,f}(t),b} \quad b, \quad \textbf{s.t.} \quad \mathbf{C1}-\mathbf{C4}, \widetilde{\mathbf{C5}}, \widetilde{\mathbf{C6}} \ \& \ \mathbf{C7},$$

$$\frac{b + \lambda\sum_{i\in\mathcal{I}}\sum_{r\in\mathcal{R}}\sum_{f\in\mathcal{F}} c_{i,r,f}(t)}{\prod_{i\in\mathcal{I}}\prod_{j\in\mathcal{J}(t)}\prod_{r\in\mathcal{R}}\prod_{f\in\mathcal{F}} \left(\frac{q_{i,j,r,f}(t)}{\chi_{i,j,r,f}(n)}\right)^{\chi_{i,j,r,f}(n)}} \le 1,$$

where the parameter $\chi_{i,j,r,f}(n)$ can be obtained by computing

$$\chi_{i,j,r,f}(n) = \frac{q_{i,j,r,f}(n-1)}{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}(t)}\sum_{r\in\mathcal{R}}\sum_{f\in\mathcal{F}} q_{i,j,r,f}(n-1)}, \quad \forall i\in\mathcal{I}, j\in\mathcal{J}(t), r\in\mathcal{R}, \forall f\in\mathcal{F}. \tag{10}$$

The problem $\widehat{\mathcal{P}}$ can be solved efficiently using CVX [14].

B. Proposed algorithm

The proposed algorithm starts with initial values for $\mathbf{I}_0$ and $\mathbf{W}_0$. By solving $\widehat{\mathcal{P}}$ iteratively, we can update $\mathbf{I}$ and $\mathbf{W}$ till all of them converge. Once we have converged values of $\mathbf{I}$ and $\mathbf{W}$, we need to round them into binary values according to the formulated constraints **C2**, **C3**, and **C4**.

C. Overhead analysis

In the current CLSPs, real time messaging protocol (RTMP) has been widely used to pull the live stream from the broadcaster and push it to the viewers. RTMP works on top of TCP, in which the message is divided into small chunks. The required information for the proposed algorithm can be piggybacked within the chunks using RTMP. Such information is only needed at most once in each time slot, which can be expressed with a few bits. Therefore, the communication overhead is trivial compared with the large amount of live video data.

IV. NUMERICAL RESULTS

A. Exhaustive search

To evaluate our algorithm, we first compare the performance of the proposed algorithm with the exhaustive search scheme by setting $I = 2$, $R = 2$, $F = 3$, and $J(t) = 3$. The performance of the proposed scheme for various λ values is collected by running the simulation 10 times and taking the average value of the results. Fig. 2 depicts the network utility achieved by both schemes versus λ, and confirms that the proposed algorithm closely approaches the globally optimal solution of transcoding task allocation and viewer association.

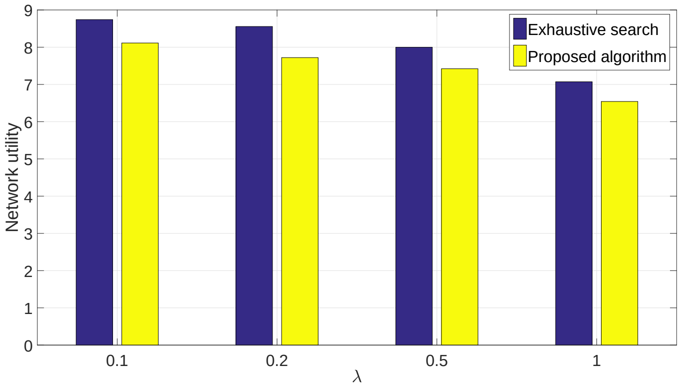

Fig. 2: Network utility versus λ

B. Trace-driven simulation

1) Simulation setup: In this section, a new simulation is built based on the real-world data of Twitch broadcaster information including the broadcaster IDs, the viewer counts and the original video resolutions [11]. According to the data set, we pick three broadcasters who are consistently online for one hour on 02/01/2015 and dynamically update the viewer counts per five minutes (which is the length of a time slot). In addition, we also find a data set containing the join and leave information of more than 7000 viewers from a chat log online [15]. For each viewer, we calculate the online time duration and estimate the online stability by normalization. Then this information is utilized as the transcoding success rate of each fog device (α_f). 23 viewers are picked to be the fog transcoders with $\alpha_f \in [0.35, 0.63]$. For the representation and the genre rate, we refer to the twitch broadcaster settings [16]. As a result, $R = 4$ and the specific bit rates of the four representations are set to be 400kbps (240P), 1500kbps (480P), 2500kbps (720P), and 5000kbps (1080P). Moreover, we define six viewer categories with the capacity set as $\mathcal{S}$={500kbps, 1000kbps, 2000kbps, 3000kbps, 4500kbps, 6000kbps} and add a bias capacity γ to each viewer capacity $s \in \mathcal{S}$ to explore the effect of the viewer capacities. In addition, we define the function $\phi(r)$ as a logarithmic function of a representation's index, namely $\phi(r) = \log(r)$. Finally, we simulate Top-N which is a currently-running cloud transcoding scheme in Twitch.TV. Top-N offers N premium broadcasters with the ABR service but only the original live video for the rest of broadcasters. To appropriately denote the cost of cloud transcoding, a constant coefficient θ is multiplied to adjust the unit cost of one transcoding task.

2) Performance evaluation: The results of this trace-driven, data-based simulation are presented in this section. In Fig. 3, the performance of the proposed scheme is plotted against the viewer counts of each broadcaster with $\lambda = 0.1$. We can find that with the number of viewers increasing, the network utility summed over all broadcasters increases. This is because the cost is paid only for each transcoding task but not for each new viewer. Therefore, when the number of viewers with multiple capacities exceeds a certain threshold, all the representations will be transcoded and the cost will stop increasing.

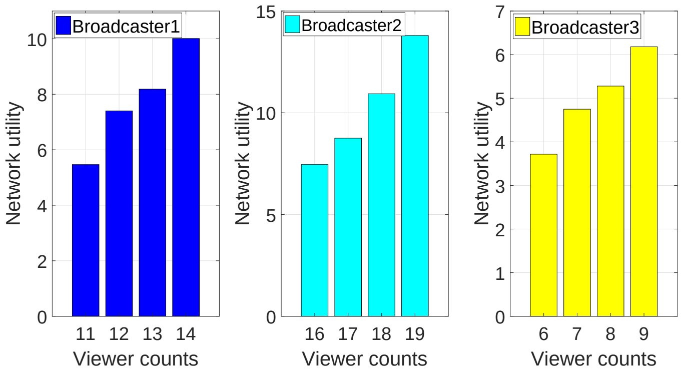

Fig. 3: Network utility versus viewer counts

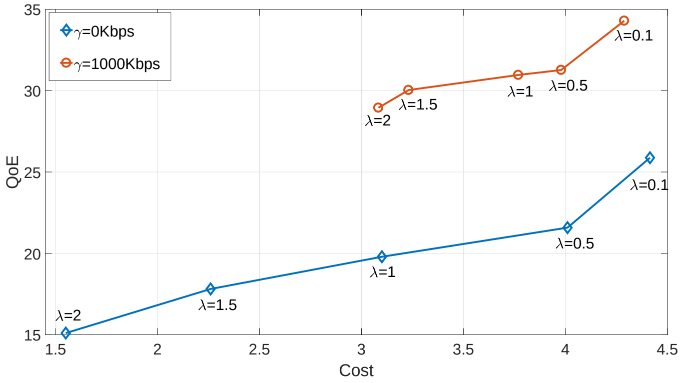

Fig. 4: Cost and QoE versus λ

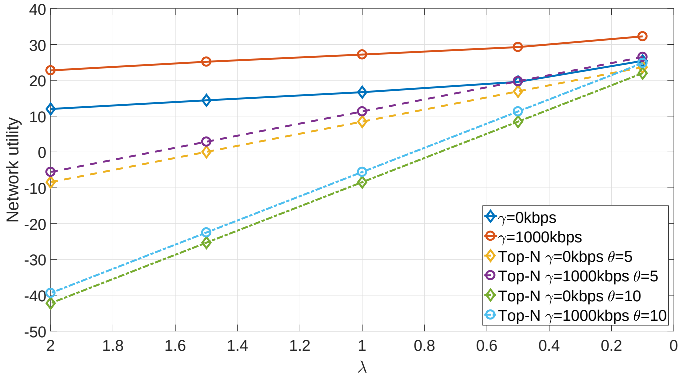

Fig. 5: Network utility versus λ

Fig. 4 depicts the total cost and QoE values versus λ with different viewer capacities. The results demonstrate that with the decrease of λ, both the QoE and the cost tend to increase. Because decreasing λ means that the cost has less impact on network utility thus more representations will be transcoded for the viewers (see Fig. 5). In addition, the higher viewer capacity will result in better QoE.

In Figs. 5 and 6, the network utility and the number of transcoded representations are plotted against λ with different viewer capacities. It is evident that both values increase with decreasing λ. The results are consistent with Fig. 4 and highlight the significance of providing the ABR service to improve the viewer QoE. In addition, Fig. 5 demonstrates that the proposed algorithm outperforms the Top-N by reaching higher network utility versus λ.

Fig. 7 describes the relationship between the number of transcoded video representations and the viewer counts. It is shown that with the increase of the viewer counts, the number of representations also increases for each broadcaster, which proves that the proposed algorithm is able to provide viewers with multiple choices of video quality versions according to the viewer capacity.

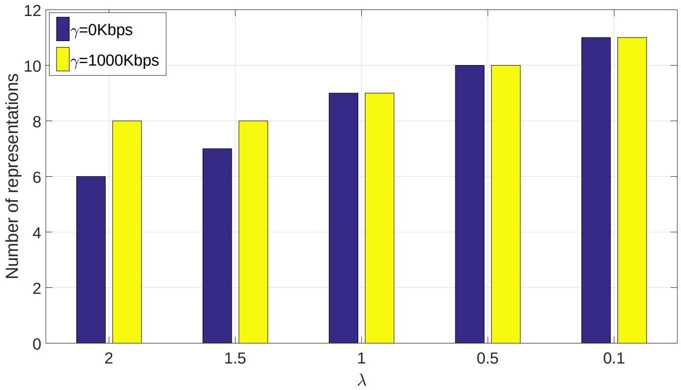

Fig. 6: Number of representations versus λ

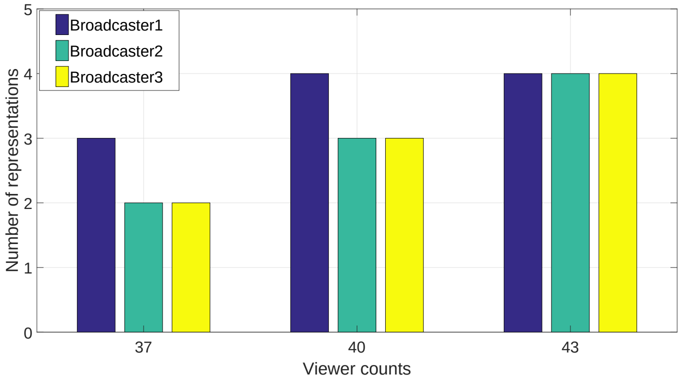

Fig. 7: Number of representations versus viewer counts

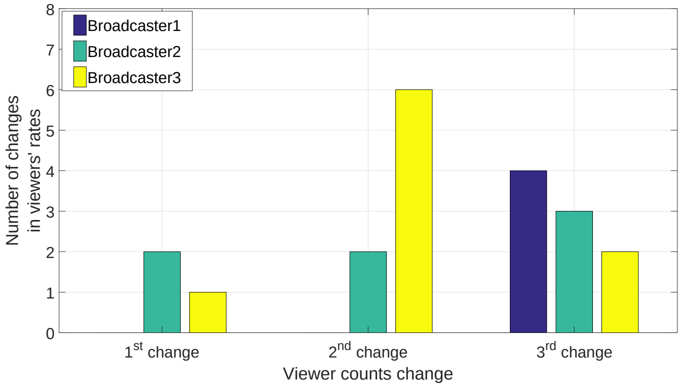

Fig. 8: Number changes in viewers' rates

Fig. 8 depicts the number of changes in viewers' rates when more viewers join (from 35 to 39, 39 to 40, and 40 to 43). With such changes, around 15 percent of viewers' experienced changes in representations on average. Although new viewers can affect the tradeoff between QoE and cost, most of the viewers receive unchanged representations, which demonstrates the proposed algorithm is able to provide the viewers with robust ABR service.

V. CONCLUSION

In this paper, a joint fog-assisted transcoding and viewer association algorithm is proposed for the CLSP. In order to maximize the network utility of the system, a non-convex integer problem is formulated and then solved by applying continuous relaxation and monomial approximations. Simulation results demonstrate that the proposed algorithm is able to find the near-optimal solution and is able to provide ABR service to the viewers.

REFERENCES

[1] K. Bilal and A. Erbad, "Impact of multiple video representations in live streaming: A cost, bandwidth, and qoe analysis," in *2017 IEEE International Conference on Cloud Engineering (IC2E)*, April 2017.
[2] Q. He, J. Liu, C. Wang, and B. Li, "Coping with heterogeneous video contributors and viewers in crowdsourced live streaming: A cloud-based approach," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 916–928, May 2016.
[3] Y. Zheng, D. Wu, Y. Ke, C. Yang, M. Chen, and G. Zhang, "Online cloud transcoding and distribution for crowdsourced live game video streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1777–1789, Aug 2017.
[4] K. Bilal, A. Erbad, and M. Hefeeda, "Crowdsourced multi-view live video streaming using cloud computing," *IEEE Access*, vol. 5, pp. 12 635–12 647, 2017.
[5] C. Zhang, J. Liu, and H. Wang, "Cloud-assisted crowdsourced livecast," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 3s, pp. 46:1–46:22, Jul. 2017. [Online]. Available: http://doi.acm.org/10.1145/3095755
[6] C. Dong, Y. Jia, H. Peng, X. Yang, and W. Wen, "A novel distribution service policy for crowdsourced live streaming in cloud platform," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 679–692, June 2018.
[7] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Video transcoding, caching, and multicast for heterogeneous networks over wireless network virtualization," *IEEE Communications Letters*, vol. 22, no. 1, pp. 141–144, Jan 2018.
[8] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Computer Networks*, vol. 130, pp. 94 – 120, 2018.
[9] K. Bilal, E. Baccour, A. Erbad, A. Mohamed, and M. Guizani, "Collaborative joint caching and transcoding in mobile edge networks," *Journal of Network and Computer Applications*, vol. 136, pp. 86 – 99, 2019.
[10] Q. He, C. Zhang, X. Ma, and J. Liu, "Fog-based transcoding for crowdsourced video livecast," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 28–33, April 2017.
[11] Q. He, C. Zhang, and J. Liu, "Crowdtranscoding: Online video transcoding with massive viewers," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1365–1375, June 2017.
[12] Y. Zhu, Q. He, J. Liu, B. Li, and Y. Hu, "When crowd meets big video data: Cloud-edge collaborative transcoding for personal livecast," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2018.
[13] M. Derakhshani, X. Wang, D. Tweed, T. Le-Ngoc, and A. Leon-Garcia, "AP-STA association control for throughput maximization in virtualized wifi networks," *IEEE Access*, vol. 6, pp. 45 034–45 050, 2018.
[14] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.
[15] Codonbyte, "Codonbyte's pastebin," https://pastebin.com/u/Codonbyte.
[16] "Broadcasting guidelines," https://stream.twitch.tv/encoding/.