First Mile in Crowdsourced Live Streaming: A Content Harvest Network Approach

Haitian Pang
Department of Computer Science and Technology, Tsinghua University
pht14@mails.tsinghua.edu.cn

Zhi Wang
Graduate School at Shenzhen, Tsinghua University
wangzhi@sz.tsinghua.edu.cn

Chen Yan
Department of Computer Science and Technology, Tsinghua University
yc.1995@foxmail.com

Qinghua Ding
Department of Computer Science and Technology, Tsinghua University
dqh14@mails.tsinghua.edu.cn

Lifeng Sun
Department of Computer Science and Technology, Tsinghua University
sunlf@tsinghua.edu.cn

ABSTRACT

Recent years have witnessed a rapid increase of crowdsourced live streaming (CLS): applications like Twitch.tv have attracted millions of daily active users. Content delivery in such crowdsourced live streaming involves two phases: 1) Video stream (i.e., a live channel) is generated and uploaded by a broadcaster user, and 2) The video stream is then delivered to many viewers choosing to watch the channel. Today's crowdsourced live streaming service usually employs conventional content delivery network (CDN) solutions to address the above content delivery problem, i.e., letting the broadcaster upload the video to a sinking CDN that then distributes the content to viewers. This solution causes a large delay and bandwidth insufficiency in the first mile between the broadcasters and the sinking CDN servers—our measurement study shows that the first-mile upload network quality causes a large portion of viewer rebuffers in the whole channel.

In this paper, we propose a content harvest network (CHN) solution to address the first-mile problem. In particular, the content harvest network employs relays at the edge of the network, to receive the content uploaded by broadcasters and then forward it to the CDN servers. Though the idea seems straightforward, it faces the following challenges: i) How to determine which channels need relay assistance? ii) How to choose the right relays to provide good first-mile QoS? iii) How to dynamically adjust the relay assignment in different channels?

In order to provide global optimal and real-time assignment, we use a hybrid solution, i.e., centralized assignment and distributed assignment. Specifically, we formulate global relay assignment as an optimization problem and develop an approximation algorithm using rounding technique. We use a multi-armed bandit (MAB) based method to perform the distributed assignment. Experiment results on a large-scale trace show that our solution can reduce the overall viewer latency by 40%, as compared to state-of-the-art solutions.

ThematicWorkshops'17, October 23–27, 2017, Mountain View, CA, USA

ISBN 978-1-4503-5416-5/17/10…$15.00
DOI: http://dx.doi.org/10.1145/3126686.3126703

CCS CONCEPTS

•Networks → Network resources allocation; Network services; Overlay and other logical network structures; •Computing methodologies → *Machine learning;* **•Computer systems organization →** *Real-time system architecture;*

KEYWORDS

Crowdsourced Live Streaming, Relay network, Broadcaster, Content Delivery Network.

1 INTRODUCTION

1.1 Background and Motivations

Cisco reports that 60% global mobile data traffic is video traffic in 2016, and forecasts that mobile video will increase 9-fold between 2016 and 2021 [1]. Among the global video traffic, CLS is an emerging live video service that has gained explosive growth these years. The key idea of CLS is that numerous widely-distributed broadcasters provide live streaming to viewers using their personal computing devices (e.g., smart phone and tablet). End users in CLS not only consume contents but also produce contents. Many real-world platforms provide CLS service to users, such as Facebook Live, Youtube Live, and Twitch.tv. A typical example is Inke.tv [2], the pioneer platform of CLS in China, of which the daily active user (DAU) reached 17 million, and the registered member reached 140 million in December 2016.

Compared to Video on Demand (VoD) services (e.g., Hulu or Netflix) and professional live streaming providers (e.g., WatchESPN), the CLS service bears larger challenge in achieving high network quality for the following reasons. The bandwidth of first-mile upload network is much less than the download link due to the asymmetry of today's network architecture. Moreover, an important characteristics of CLS is the real-time interaction between broadcasters and viewers, hence requires ultra low network transmission latency. In addition, the massive broadcasters establish and close broadcast channels dynamically, incurring high churn rate of the source streamings.

Due to the above challenges, the currently wide-adopted CDN architecture fails to support the massive and dynamic uploading sources in the CLS service efficiently. Previous work finds that the streaming service is vulnerable and sensitive when the broadcaster's networking capacity changes frequently, and this will even cause viewers quit the channel [3]. The first-mile network problem will be more severe considering that most broadcasters employ mobile devices and mobile networks for live streaming.

In this way, novel architecture should be designed to improve the performance of the CLS service. To this end, we design a relay-assisted content harvest network for CLS service, using relays to upload broadcaster streaming to servers. With the advent of emerging technologies such as edge computing [4] and fog computing [5], edge nodes are playing an important role in content delivery [6]. The key idea of CHN is to utilize dedicated edge nodes to form a relay network, providing the broadcasters with optional routing schemes for lower network cost. Specifically, a live streaming can be delivered directly to the server or relayed to the relays first and then delivered to the server. The challenges for this novel architecture are as follows:

- How to determine which channels need relay assistance.
- How to choose the right relays to provide good first-mile QoS for the broadcasters.
- How to dynamically adjust the relay assignment in different channels as the network condition evolves over time.

To address the above challenges, we design a hybrid relay assignment solution for the CLS service. Specifically, we employ centralized assignment to provide global optimality using an optimization method in a periodic way, and distributed assignment to provide fast response using multi-armed bandit based method when the channel establishes.

The contributions of the paper are multi-fold:

- We design a novel architecture CHN to help provide relays for first-mile network aiming to reduce the loss rate and latency of viewers in CLS.
- We propose a hybrid (centralized and distributed) solution to assign the relay. Specifically, we formulate the centralized relay assignment problem as an optimization problem, prove it is NP-hard, and design an algorithm to solve the problem in polynomial time. To provide real-time response, we enable edge devices to perform relay assignment in a distributed way, and employ MAB-based method to solve the problem.
- Experimental results on a large scale dataset of a typical CLS platform show that with the aid of CHN, the total viewer cost reduces by 40% compared to conventional CDN-based method. The MAB-based method can reduce latency by 27% compared to static assignment.

The remainder of this paper is organized as follows. Section 2 introduce the related work. Section 3 introduce the architecture design. Section 4 present the centralized relay assignment method. Section 5 present the distributed relay assignment method. We show experiment results in Section 6, and conclude in Section 7.

2 RELATED WORK

2.1 Live Streaming System Optimization

With the growing popularity among users, CLS has also attracted attention from the academia. [7, 8] conducted measurement study on live streaming systems like Meerkat and Periscope, and found that nowadays the typical live streaming service employs CDN to deliver contents. Many previous works focus on the optimization of the live streaming system. Mukerjee et al. [9] designed a hybrid controller to improve the performance of the CDN, and further used a centralized algorithm for live video optimization. Chen et al. [10, 11] presented a generic framework that facilitates a cost-effective cloud service for crowdsourced live streaming. Wang et al. [12] proposed CALMS, which adaptively leases and adjusts cloud server resources. He et al. [13] introduced a framework that utilized cloud computing services to enhance the viewer satisfaction and allocate the geo-distributed computing resources.

Most above mentioned works on live streaming focus on the optimization of the cloud network [11, 13], and the intra-CDN link [9]. None of the works focus on the optimization of the first-mile upload network. While, measurement study in [3] showed that the dynamic uploading capacity of broadcasters is a critical challenge, which noticeably affects the smoothness for viewers. This inspires us to design novel architecture to improve the first-mile network.

2.2 Application of relay network

Relay network has been applied in many scenarios, such as virtual private networks (VPNs) and multicast. Savage et al. [14] found that in 30-80% of the cases, there is an alternate path with significantly superior quality. During last few years, relay network has been used in novel network applications like Internet Telephony and live streaming. Jiang et al. [15] presented overlay network to improve the quality of internet telephony call. They used data analysis and machine learning methods to probe the network condition and schedule the relay decision. Zhang et al. [16] designed a data-driven overlay network to improve live streaming quality. They use the relay network in the P2P and server-client scenarios. However with the advent of CLS, employing the relay network to optimize the first-mile upload network in CLS system is more important.

3 ARCHITECTURE DESIGN

3.1 Overview of CDN-based CLS

We first introduce a typical CDN-based CLS architecture in Fig. 1, which is adopted by many CLS providers like Periscope and Twitch. In level-1, the widely distributed broadcasters upload video streaming to the upload server (upServer) of the CDN (level-2). These upServers are geo-distributed, so that broadcasters can choose a cost-efficient server for video distribution. Furthermore, the upServer will transcode the video to multiple quality versions, which is a computation-intensive. Then the transcoded streaming will be delivered to the download servers (level-3). The download servers are distributed in different geographical locations (e.g., U.S. West and China East), and will server the viewer requests (level-4) in its region.

The first-mile network improvement between level-1 and level-2 is necessary [3]. Therefore, we pay particular attention to the broadcaster upload network, and aim to use the relay network to enhance the performance of the upload network with the proposed CHN approach.

3.2 Introduction to Content Harvest Network

We present the architecture of CHN in Fig. 2, where we incorporate relays into first-mile network. Each upload streaming can take either the "direct path" (blue arrow) or the "relayed path" (red arrow).

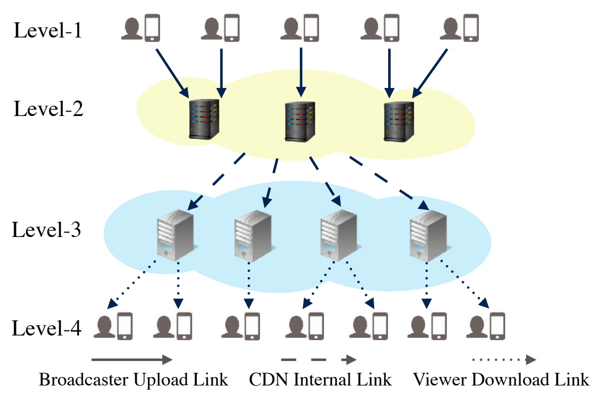

Figure 1: CDN-based live streaming system.

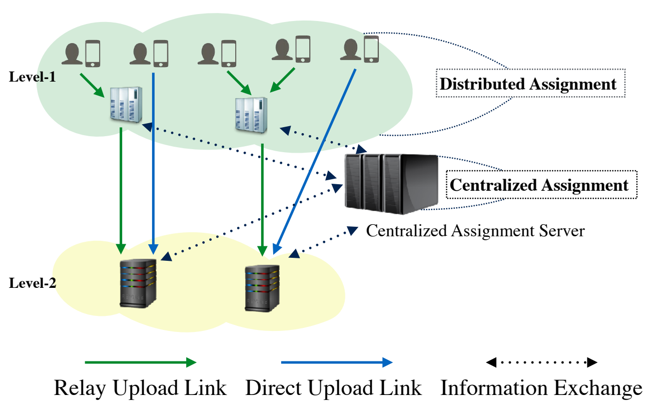

Figure 2: Architecture of content harvest network.

When taking a "relayed path", the streaming is first forwarded to a relay, and then forwarded to an upServer. The relays are provided by some fog network operator, who crowdsource the resource of edge devices [4, 6].

The relay assignment problem is based on the network performance. Specifically, the network condition between each relay-upServer pair, broadcaster-relay pair and broadcaster-upServer pair is measured and transfer to the centralized assignment server. Keeping track of the network property between the relay and upServer is feasible, as both are controlled by the CLS provider. However, the network properties of broadcasters (e.g., IP address, AS) are highly dynamic and large in number, making direct measurement infeasible. Previous works [15] employ data-driven approaches such as cluster methods to predict the network peroperties. This is a well-studied topic and is out of scope in this work. With the network performance collection from direct measurement and prediction, the network condition keeps up to date, which can be used for relay assignment.

Given the system architecture above, we present the design principles as follows:

- Hybrid Solution: We implement centralized assignment to optimize the relay assignment problem of all the broadcasters periodically. In addition, distributed assignment is required to cope with the relay assignment problem in real-time when one specific broadcast channel establishes.
- Prioritized broadcast channel: Different channels serve different number of viewers, and the popular channels should be prioritized for relay selection to benefit more viewers.
- Cost-efficient path: The relay path should be selected optimally to reduce the streaming delivery cost in the path.

3.3 Hybrid assignment

We further introduce the hybrid assignment solution in details, which determine the relay assignment at different time scales and broadcaster number scales. The hybrid solution determines whether a broadcast streaming should be relayed, and further which relay to use. The centralized assignment takes the whole network information as input, and calculates the optimal relay assignment of all broadcasters as output. Due to large computation, the centralized assignment operates in a periodic way (e.g., several minutes). We formulate the centralized assignment problem as an optimization problem and provide efficient approximation algorithm in Section 4. The distributed assignment makes quick decision for a better network condition in subsecond response time when a broadcast channel establishes. The distributed assignment makes this decision using MAB method based on historical data, and the algorithm is provided in Section 5.

3.4 An illustrative example

To introduce the architecture features and demonstrate the key design principles (prioritized broadcast channel and cost-efficient path), we provide an illustrative example in Fig. 3, where we consider a system with two broadcasters $\{B_1, B_2\}$, two relays $\{R_1, R_2\}$, and one upServer $\{U\}$. The bitrates of the broadcasters are $[800, 400]$ kbps. The capacity of the relay-upServer can be measured, which are $[1000, 800]$ kbps. The link cost of each node pair is denoted in Fig. 3, representing the QoS loss caused by packet loss and transmission delay. Since both the loss rate and delay are additive, we can derive the path cost table in Table 1.

-	ϕ	R_1	R_2
B_1	10	7	8
B_2	11	2	6

Table 1: Relay Cost Matrix.

We show the optimal relay assignment policy in Fig. 3, where the broadcaster popularities are $[1000, 10]$ and $[10, 10]$, respectively. When the popularity of B_1 is very large, it gets the priority to select the most cost-efficient path $[B_1, R_1, U]$, although B_2 can obtain the larger cost degradation when selecting R_1 as the relay. When the popularity of B_1 and B_2 are the same, B_2 is assigned to R_1, as the path cost is reduced most remarkably.

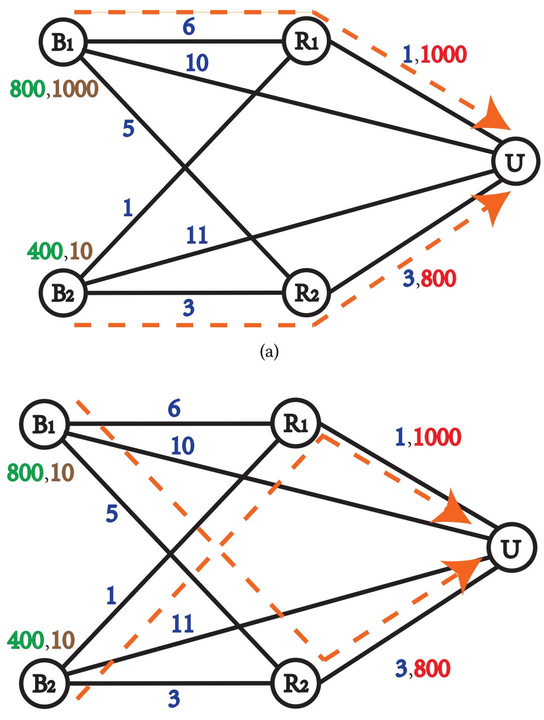

Figure 3: An illustrative example of relay assignment optimization.

4 CENTRALIZED ASSIGNMENT: FORMULATION AND OPTIMIZATION

In this section, we provide the formulation of the global relay assignment problem, prove it is NP-hard and design fast algorithms to achieve sub-optimal performance. As centralized assignment in computation-intensive, it works in a periodic way.

4.1 Basic Network Model

The broadcaster set can be denoted as $\mathcal{B} = \{1, 2, ..., B\}$, which is the source of the first-mile network. The edge relays, i.e., devices with network transmission capacity, are used to relay data from the broadcasters to the upServer, and we denote the relay set as $\mathcal{R} = \{1, 2, ..., R\}$. The destination of the broadcaster streams is the upServers denoted as $\mathcal{U} = \{1, 2, ..., U\}$. We assume that the upServer has the limit of computation, e.g., transcoding, and we denote the limit as *Com*, which is measured in *bps*. As we allow one-hop relay, the broadcaster will be assigned to one relay or no relay, depending on the relay assignment. The relay assignment option of each broadcaster b is denoted as $r(b) \in \mathcal{R} \cup \{0\}$, where $r(b) = 0$ means that the broadcaster is assigned to no relays, and delivers the stream to the upServer directly. Thus, we can denote the relay option set as $\mathcal{R}^* = \mathcal{R} \cup \{0\} = \{0, 1, 2, ..., R\}$. Whether or not the broadcaster is assigned to a relay, the destination of the broadcaster stream is one specific upServer of the CDN. Specifically, the upServer assignment of each broadcaster b is denoted as $u(b) \in \mathcal{U}$. The centralized assignment aims to determine the relay and upServer assignment $(r(b), u(b)); \forall b \in \mathcal{B}$ to optimize the global system performance.

4.2 Network Cost Model

The relay assignment problem relies on the network quality of the broadcaster-to-relay link, the broadcaster-to-upServer link, and the relay-to-upServer link. We denote the link capacity between relay r and upServer u as $c(r, u)$. Hence, $C^{R-U} = \{c(r, u)\}; \forall r \in \mathcal{R}, u \in \mathcal{U}$ is a matrix representing the link capacities of all possible relay-to-upServer pairs. Note that we cannot derive the link capacity related to the broadcaster, as the broadcaster is not under control and the bandwidth is highly dynamic. However, we can estimate the loss rate and delay between any node pair of the uploading path, as mentioned in [15]. For a node pair (i, j) in the uploading path, the loss rate is denoted as $l(i, j)$, and the delay is denoted as $d(i, j)$. Based on above definitions, we define the video QoS of the node pair as a weighted sum of loss rate and delay measured [19–21]. The link cost $s_{i,j}$ can be formulated as follows:

$$s_{ij} = \alpha d(i, j) + (1 - \alpha)l(i, j),\ 0 < \alpha < 1,\ \forall i, j \in \mathcal{B} \cup \mathcal{R} \cup \mathcal{U}. \quad (1)$$

The path cost is defined as $S(b, r(b), u(b))$, reflecting the loss rate and transmission latency of the link:

$$S(b, r(b), u(b)) = \begin{cases} s(b, u(b)) & r(b) = 0 \\ s(b, r(b)) + s(r(b), u(b)) & r(b) \in \mathcal{R}^*/\{0\} \end{cases} \quad (2)$$

4.3 Broadcaster Model

The bitrate of each broadcaster is different, as each broadcaster has unique device setting and video content[1]. We denote the bitrate of broadcaster b as $t(b)$. Based on the analysis about the broadcaster patterns, we notice that the viewer number varies among different broadcasters, which inspires us to illustrate the popularity of broadcasters. We define the current viewer number as $P_c(b)$, which is a measure of the channel popularity. However, the centralized assignment works in a periodic way, thus requires future information. Hence, a more accurate popularity method is required. In this paper, we develop a technique to estimate the popularity of a channel by focusing on the weighted viewer number of current viewer number and broadcaster average viewer number. To predict the viewer number, we use exponential moving average as follows:

$$P(b) = (1 - \beta)P_a(b) + \beta P_c(b),\ 0 < \beta < 1, \quad (3)$$

where $P(b)$ is the popularity of the broadcaster b, $P_a(b)$ is the average concurrent viewer number of broadcaster b, β is the smoothing factor. Larger values β reduces the smoothing level, and when $\beta = 1$ the popularity is the current viewer number. Thus, the overall viewers' cost of channel b is $P(b) \cdot S(b, r(b), u(b))$. The overall cost of the viewers in the system Q is formulated as follows:

$$Q = \sum_{b \in \mathcal{B}} P(b) \cdot S(b, r(b), u(b)) \quad (4)$$

4.4 Problem Formulation

Our aim is to minimize the overall cost of all the viewers by assigning the relay decisions in the first-mile network. We introduce $x_b(r, u) \in \{0, 1\}, \forall r \in \mathcal{R}^*, \forall u \in \mathcal{U}$ to represent the joint relay and upServer assignment, $\vec{x}_b$ follow the constraint:

$$|\vec{x}_b| = 1, \forall b \in \mathcal{B}, \quad (5)$$

[1] [13] detected that the bitrates in a CLS platform are highly heterogeneous.

which indicates that there is only one path from the broadcaster to the upServer.

As the relay-to-upServer data traffic cannot exceed the link capacity, we have the following link capacity constraints:

$$\sum_{b \in \mathcal{B}} t(b) \cdot x_b(r,u) \leq l(r,u), \quad \forall r \in \mathcal{R}, u \in \mathcal{U}, \tag{6}$$

The upServer is computation intense as the video transcoding and forwarding is performed in it. As [22] illustrated that the more popular videos should be transcoded into more replications, we define $H(P(b))$ as a concave increasing function of the computing resource spent when transcoding the streaming of broadcaster b with popularity $P(b)$. Hence, the upServer computing constraint can be denoted as:

$$\sum_{b \in \mathcal{B}} t(b) H(P(b)) \cdot \sum_{r \in \mathcal{R}^*} x_b(r,u) \leq Com, \quad \forall u \in \mathcal{U}, \tag{7}$$

Then $S(b, r(b), u(b))$ can be reformulated as $S^*(b, \vec{x}_b)$, which is shown as follows:

$$S^*(b, \vec{x}_b) = \sum_{r \in \mathcal{R}^*} \sum_{u \in \mathcal{U}} S(b,r,u) \cdot x_b(r,u) \tag{8}$$

The global relay assignment problem can be formulated as:

$$\begin{aligned} \min \sum_{b \in \mathcal{B}} \sum_{r \in \mathcal{R}^*} \sum_{u \in \mathcal{U}} & P(b) \cdot S(b,r,u) \cdot x_b(r,u) \\ & \text{subject to } (5), (6), (7) \\ x_b(r,u) \in \{0,1\}, & \forall b \in \mathcal{B}, r \in \mathcal{R}^*, u \in \mathcal{U}, \end{aligned} \tag{9}$$

We can achieve the optimal relay assignment via solving the above problem, and we define the problem as optimal assignment problem (OAP).

***Theorem** 1.* The optimal assignment problem (OAP) is NP-hard.

4.5 Algorithms

Since OAP is proved to be NP-hard, we cannot obtain the optimal solution in polynomial time. In this section, we provide Greedy Rounding Algorithm (GRA) for the OAP with rounding technique, and prove that the method is theoretically bounded. GRA method runs relatively fast in small and medium network scale, while in large scale network, we need faster implementation. In order to accelerate the calculation process, we further develop a fast implementation (FGRA) of GRA, which has computing complexity in polynomial time and fits the large-scale network.

4.5.1 ***Greedy Rounding Algorithm***. Intuitively, we prioritize the optimization of popular broadcasters to get larger profit. But direct greedy strategy may not provide satisfying results as it ignores the network quality. Thus more consideration is needed to design a better strategy. Motivated by the rounding techniques [23], we design a greedy rounding algorithm (GRA). First we relax the binary variant constraints as below.

$$x_b(r,u) \geq 0, \forall b \in \mathcal{B}, r \in \mathcal{R}^*, u \in \mathcal{U} \tag{10}$$

This relaxation then changes the original problem into a linear programming problem. This relaxed linear programming can be effectively solved using classical methods like simplex method [24]. Suppose the result of the linear programming is $x_b^*(r,u)$, we further need to derive a feasible integer solution. Here we use the cost brought by this solution as its weight.

$$W(x_b^*(r,u)) = P(b) \cdot S(b,r,u) x_b^*(r,u) \tag{11}$$

We round the broadcaster in weight descending order. Furthermore, in order to satisfy constraint 5, we can only round the $x_b(r,u)$ with the highest weight. That is, we round according to the following policy.

$$x_b(r,u) = \begin{cases} 0 & \text{if } (r,u) \neq \arg_{(r,u)} \max W(x_b^*(r,u)) \\ 1 & \text{if } (r,u) = \arg_{(r,u)} \max W(x_b^*(r,u)) \end{cases} \tag{12}$$

Up to now, the link capacity and computation constraints may still be violated. Thus we cannot directly round the term with the highest weight, but check the first feasible rounding term and them round it to one. We show the greedy rounding algorithm in Algorithm 1.

Algorithm 1 Greedy Rounding Algorithm

procedure GREEDY-ROUNDING($\boldsymbol{P}, \boldsymbol{l}, \boldsymbol{S}, \boldsymbol{t}, Com$)
    I. Relaxation
    Solve the relaxed LP and get fraction results $x_b^*(r,u)$
    II. Rounding
    for $b \in \mathcal{B}$ ranked by $\sum_{r,u} P(b)S(b,r,u)x_b^*(r,u)$ **do**
        Rank the $(r,u), r \in \mathcal{R}^*, u \in \mathcal{U}$ by $W(x_b^*(r,u))$
        $(r_b, u_b) \leftarrow$ the first (r,u) that satisfies the const.
        $x_b(r_b, u_b) \leftarrow 1$
        for $(r,u) \neq (r_b, u_b), r \in \mathcal{R}^*, u \in \mathcal{U}$ **do**
            $x_b(r,u) \leftarrow 0$
        end for
    end for
end procedure

4.5.2 ***Fast Implementation for Greedy Rounding Algorithm (FGRA)***. We notice that in GRA, we need to solve a linear programming in the relaxation part. The most widely-accepted method is simplex method, which may induce exponential complexity [24] when the network scale gets large. As a result, we develop FGRA, which uses a heuristic to approximate GRA method. We look into $x_b^*(r,u)$ in the relaxed linear programming problem, and find that the path with lower cost corresponds to a larger $x_b^*(r,u)$. Heuristically, we want to find approximation for $x_b^*(r,u)$. We define γ as the lowest achieved cost without relay:

$$\gamma = \min_u (S(b, R+1, u)), \forall u \in \mathcal{U} \tag{13}$$

Then we use heuristic to define $x_b^*(r,u)$:

$$x_b^*(r,u) = e^{\gamma - S(b,r,u)}, \forall b \in \mathcal{B}, r \in \mathcal{R}^*, u \in \mathcal{U}. \tag{14}$$

After obtaining $x_b^*(r,u)$, the rounding process in FGRA method is the same as the GRA method. We have the following theorem for the complexity of the heuristic greedy rounding algorithm(FGRA).

***Theorem** 2.* The FGRA has polynomial time complexity.

5 DISTRIBUTED ASSIGNMENT: MAB-BASED METHOD

As the dynamic of broadcasters is frequent, thus real-time assignment is necessary for quick decision making when a broadcast channel is established. Due to time-consuming computation of global optimization, the centralized assignment is not appropriate for fast response of a newly established channel. In this way, we design a distributed assignment method for broadcasters when he establishes the broadcast channel. The assignment is made by the broadcast device and relays in a distributed two-phase manner. In phase I, the broadcaster uses MAB-based method for relay assignment based on the historical network performance. In phase II, the relay forwards the streaming to the optimal upServer based on instant relay load.

5.1 Phase I: broadcaster local assignment

For a particular broadcaster b, our goal is to assign a relay option with the lowest cost, i.e., the broadcaster can use the default path or a relayed path. Recall that R^* is the set of relay options and $s(b, r)$ is the expected network cost when using r as the relay option. The goal is to reach a dynamic balance between exploring suboptimal decision and exploiting currently optimal decision. Based on the historical relay assignment data, an *exploration and exploitation* based method can solve the relay assignment problem. Specifically, we formulate the above problem as a multi-armed bandit problem, where each relay option is a "bandit arm" and the network performance is the "reward". Our goal is to minimize the network cost with historical sessions.

Algorithm 2 shows our approach. We choose the multi-armed bandit with Upper Confidence Bound (UCB) action selection [25] as our basic method, because the UCB method considers both how close the choice is to optimality and the skewness historical selection. A represents the chosen relay node, Q denotes the estimated action value, and N denotes the number that a relay node has been chosen before. We make one modification to the basic algorithm in order to make it work well in our context. Classical MAB problem maximizes the expected reward. As the network performance we collect is network cost, we minimize the network metric $s(b, r)$ in our context.

Algorithm 2 Multi-armed bandit with Upper-Confidence-Bound

procedure MAB-UCB(b, R^*, A)
  I. Initialization
  $t \leftarrow 1$
  for $r = 1$ to R^* **do**
    $Q(r) \leftarrow 0$
    $N(r) \leftarrow 0$
  end for
  II. Recursion
  $A \leftarrow \arg\min_r Q(r) + c\sqrt{\frac{\log t}{N(r)}}$
  $R \leftarrow bandit(A)$
  $t \leftarrow t + 1$
  $N(A) \leftarrow N(A) + 1$
  $Q(A) \leftarrow Q(A) + \frac{1}{N(A)}(R - Q(A))$
end procedure

5.2 Phase II: relay local assignment

In Phase I, a newly launched broadcast channel will be relayed directly to the upServer or the relay. Once a channel is forwarded to a relay, a further decision should be made as to which upServer to assign. This optimal assignment is made by the relay considering the link capacity and load. The relay can obtain the optimal solution of upServer selection. The relay can filter all available links whose capacity is larger than the streaming bitrate and choose the link with the best network performance.

6 EXPERIMENT RESULT

6.1 Experiment Setup

We conduct the experiments on the real-world dataset collected by Inke.tv from December 9, 2016 to February 27, 2017. Each broadcaster entry corresponds to one established channel, which contains broadcaster ID, channel ID, timestamp of the channel establishing and closing, and the broadcaster location (in longitude and latitude). Each viewer entry corresponds to one view session in a broadcast channel, containing viewer ID, channel ID, IP address, timestamp of viewers entering and exiting, network connection type (e.g., WiFi or 4G) and the viewer location (in longitude and latitude). We select 1000 broadcasters from the data trace, i.e., $B = 1000$, whose bitrates range from 1Mbps to 3Mbps. We set $U = 4$ as the upServer number. We assume that the computing limitation of each upServer is 2,000 Mbps. We further set the default relay number as 100, i.e., $R = 100$, and the default α as 0.4.

We implement five methods of centralized assignment for comparison, i.e., VDN-C, TOP-N, GRA, FGRA, OPT. VDN-C [9] is a centralized assignment method to allocate resource optimally in live streaming system without relay network, and serves as baseline. TOP-N uses the relay network and assigns the broadcasters sequentially by the popularity of channels. TOP-N also serves as the baseline, which is an intuitive assignment method for the relay assignment. OPT is the theoretically optimal solution of the relay assignment problem. We implement three distributed methods, i.e., Static, Greedy, UCB. The "Static" method chooses the same relay all the time. The "Greedy" method selects the currently optimal relay based on historical data.

6.2 Trace-Driven Results

We now present the viewers' normalized costs under five methods in Fig. 4. We notice that with the implement of relay network, the viewer cost can be reduced by 25% ~ 43%, as the VDN-C method has the highest cost, delay and loss rate. The OPT method has the lowest cost, and is regarded as the optimal baseline. TOP-N method induces the highest cost in relay-based methods. The costs of GRA and FGRA are very close, and the cost of FGRA is slightly higher that GRA. This suggests that the heuristics do not cause much accuracy loss. Compared with OPT, GRA incurs only 3% more cost, and FGRA incurs 8% more cost. Meanwhile, TOP-N incurs 25% more cost than the OPT method.

We show the viewer cost versus relay number in Fig. 5, where we randomly remove some relays and calculate the viewer cost. We find the viewer cost reduce with relay number for the relay based methods.

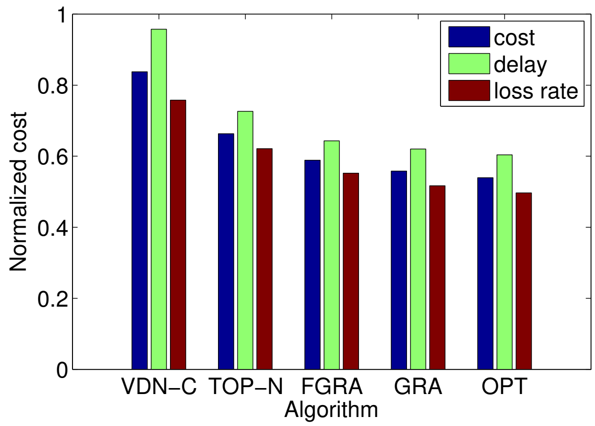

Figure 4: Normalized Cost of different algorithms.

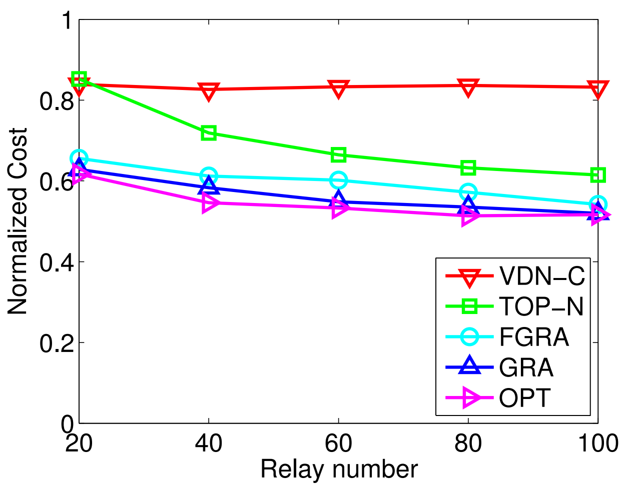

Figure 5: Normalized Cost versus relay number.

We show the total cost as a function of the refresh period of the centralized assignment. Table. 2 show the total cost with the deployment of only the centralized assignment under different refresh period. For fair comparison, the total cost is calculated in a day based on the same set of broadcast and viewship. We observe that the finer granularity results in higher cost reduction. Note that in reality, the period should be set considering the trade-off between the cost reduction and the computation overhead.

We show the performance under different iteration times of different distributed assignment algorithms in Fig. 6 and Fig. 7. We find that our MAB-based method outperforms the baselines greatly, and can select the path with higher link capacity (140% more than baseline) and lower latency (27% lower than baseline).

Period	5min	10min	15min	20min	30min	1hour
Cost ($\times 10^7$)	3.19	3.89	4.07	4.68	5.31	7.26

Table 2: Total Cost versus Period of Centralized Assignment.

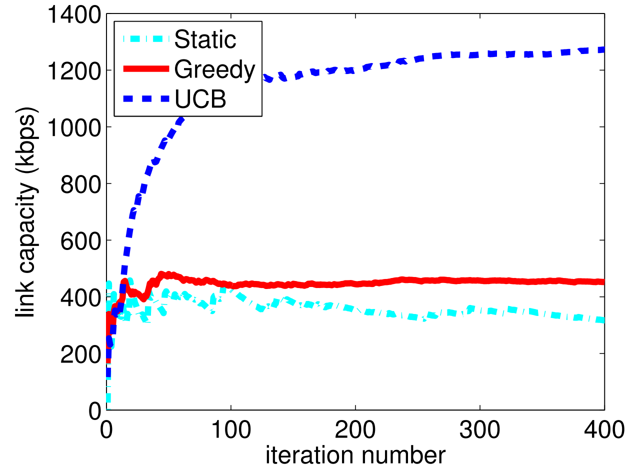

Figure 6: Distributed assignment methods to maximize link capacity.

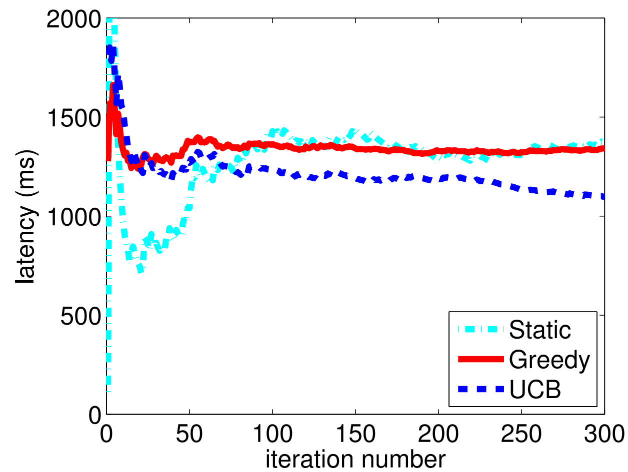

Figure 7: Distributed assignment methods to minimize latency.

We further investigate the total cost in a typical day with different methods in Fig. 8. For fairness comparison, we set the refresh period as 30 minutes for all methods. As the VDN-C method do not utilize the relay network, the VDN-C method induces the highest cost of all time and serves as benchmark. Deploying only the Centralized assignment method (we choose FGRA for practical

Relay Number	Processing Time (sec)			
	TOP-N	FGRA	GRA	OPT
200	0.05	0.12	69.02	1872.76
400	0.09	0.22	146.84	3640.71
600	0.13	0.34	235.96	NA
800	0.17	0.49	340.11	NA
1000	0.21	0.59	455.69	NA

Table 3: Processing time versus relay number.

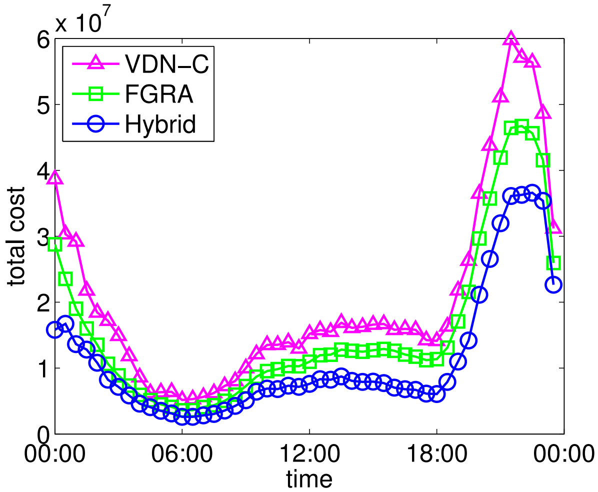

Figure 8: Total Cost in a day.

consideration) induces lower cost than VDN-C. The hybrid assignment method utilizing centralized method (FGRA) and distributed method (UCB) achieves the best performance. Comparing with the pure FGRA, the gap of cost enlarges during peak demand and reaches 20% cost reduction. This is because in peak demands, the dynamic of broadcast channels increase, thus more new-launched broadcast channels will be assigned by distributed assignment. The hybrid method outperforms the VDN-C method by 40% in cost reduction.

Above results indicate the relay network is notable for reducing the cost. As a real-world on-line implementation, another important metric is the time consumption in scheduling. On one hand, the network condition shifts quickly, thus scheduling should be refreshed in a few minutes. On the other hand, the number of broadcasters and relays may be very large in reality, challenging the processing speed. We present the processing time of different methods with the number of relay and broadcaster in Table. 3 and Table.4, respectively. We find that the processing time of FGRA is 1000x faster than GRA. It takes about 6 minutes to finish the assignment process when the broadcaster number reaches 100, 000. The running time measurement was conducted on a desktop with Intel Core i7-6700 CPU @ 2.6 GHz x 4.

7 CONCLUSION

In this paper, we present a relay-assistant network for content harvesting in first-mile network for crowdsourced live streaming. We use a hybrid solution, i.e., joint centralized and distributed assignment, to perform the relay assignment. We model the centralized relay assignment as an optimization problem, and developed an optimal algorithm and a fast approximation algorithm. We utilize the MAB-based method to perform the distributed assignment locally. The performance of the proposed solution is evaluated through extensive experiments.

Broadcaster Number	Processing Time (sec)			
	TOP-N	FGRA	GRA	OPT
10^1	0.02	0.03	8.87	17.13
10^2	0.12	0.29	94.22	192.50
10^3	1.16	3.56	1001.32	2133.34
10^4	13.25	37.75	NA	NA
10^5	138.53	406.33	NA	NA

Table 4: Processing time versus broadcaster number.

8 ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61472204, and 61521002, Beijing Key Laboratory of Networked Multimedia under Grant No. Z161100005016051, and the Technology Plan of Shenzhen under Grant No. CXZZ20150323151850088.

A PROOF OF THEOREM 1

We prove the NP-hardness of OAP by reduction to the knapsack problem. Suppose there is only one relay node and one upload server in this broadcast system, i.e. $\mathcal{R}^* = \{0, 1\}$, $\mathcal{U} = \{1\}$. And we assume the computation capacity of server is unlimited, i.e., $Com = +\infty$, thus the constraints in equation 7 always satisfies. Furthermore, we assume that all broadcasters taking the relay path bear the same link quality and we normalize it to 1, i.e., $S(b, 1, 1) = 1, \forall b \in \mathcal{B}$ and $S(b, 0, 1) = 0, \forall b \in \mathcal{B}, \forall u \in \mathcal{U}$. Moreover, the constraint of capacity on path $\{1, 1\}$ is denoted as $l(1, 1) = W$. The problem becomes "How to maximize the viewer number transmitted via relay network under the constraint of the path capacity, given that each broadcaster bears a bitrate and a viewer number." Specifically, the original problem is reduced to the following problem:

$$\begin{aligned} \max \sum_{b \in \mathcal{B}} P(b) x_b \\ \sum_{b \in \mathcal{B}} t(b) x_b \leq W \\ x_b \in \{0, 1\}, \forall b \in \mathcal{B}, \end{aligned} \tag{15}$$

We notice that the above problem is a classical 0-1 knapsack problem. As the 0-1 knapsack problem is known to be NP-hard [26], we prove the NP-hardness of the OAP.

B PROOF OF THEOREM 2

First we consider the complexity of rounding process - the second part in the algorithm. The time limit appears at the step of ranking the (r, u) pairs for each $b \in \mathcal{B}$. The complexity of this rounding part is of $O(|\mathcal{B}| \cdot |\mathcal{U}| \cdot |\mathcal{R}| \log(|\mathcal{U}| \cdot |\mathcal{R}|))$, which is polynomial.

REFERENCES

[1] Cisco visual networking index: Global mobile data traffic forecast update 20170209
[2] http://www.inke.tv/
[3] Zhang C, Liu J. On crowdsourced interactive live streaming: a twitch. tv-based measurement study, ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, 2015.
[4] Shi W, Cao J, Zhang Q, et al. Edge computing: Vision and challenges, IEEE Internet of Things Journal, 2016.
[5] Chiang M. Fog networking: An overview on research opportunities. arXiv preprint arXiv:1601.00835, 2016.
[6] Chen L, Zhou Y, Jing M, et al. Thunder crystal: a novel crowdsourcing-based content distribution platform, ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, 2015.
[7] Wang B, Zhang X, Wang G, et al. Anatomy of a personalized livestreaming system, ACM Internet Measurement Conference, 2016.
[8] Siekkinen M, Masala E, Kämäräinen T. A First Look at Quality of Mobile Live Streaming Experience: the Case of Periscope, ACM on Internet Measurement Conference, 2016.
[9] Mukerjee M K, Naylor D, Jiang J, et al. Practical, real-time centralized control for cdn-based live video delivery, ACM SIGCOMM Computer Communication Review, 2015.
[10] Chen F, Zhang C, Wang F, et al. Cloud-Assisted Live Streaming for Crowdsourced Multimedia Content, IEEE Transactions on Multimedia, 2015.
[11] Chen F, Zhang C, Wang F, et al. Crowdsourced live streaming over the cloud, IEEE Conference on Computer Communications (INFOCOM), 2015.
[12] Wang F, Liu J, Chen M, et al. Migration towards cloud-assisted live media streaming, IEEE/ACM Transactions on networking, 2016.
[13] He Q, Liu J, Wang C, et al. Coping with heterogeneous video contributors and viewers in crowdsourced live streaming: A cloud-based approach, IEEE Transactions on Multimedia, 2016.
[14] Savage S, Collins A, Hoffman E, et al. The end-to-end effects of Internet path selection, ACM SIGCOMM Computer Communication Review, 1999.
[15] Jiang J, Das R, Ananthanarayanan G, et al. Via: Improving internet telephony call quality using predictive relay selection, ACM SIGCOMM, 2016.
[16] Zhang X, Liu J, Li B, et al. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming, IEEE Conference on Computer Communications (INFOCOM), 2005.
[17] Qiu T, Ge Z, Lee S, et al. Modeling channel popularity dynamics in a large IPTV system, ACM SIGMETRICS Performance Evaluation Review, 2009.
[18] Madhyastha H V, Isdal T, Piatek M, et al. iPlane: An information plane for distributed services, Symposium on Operating systems design and implementation. USENIX Association, 2006.
[19] Mok R K P, Chan E W W, Chang R K C. Measuring the quality of experience of HTTP video streaming, IFIP/IEEE International Symposium on Integrated Network Management (IM), 2011.
[20] Egilmez H E, Civanlar S, Tekalp A M. An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks, IEEE Transactions on Multimedia, 2013.
[21] He Q, Zhang C, Liu J. Utilizing Massive Viewers for Video Transcoding in Crowdsourced Live Streaming, IEEE International Conference on Cloud Computing (CLOUD), 2016.
[22] Wang Z, Sun L, Wu C, et al. Joint online transcoding and geo-distributed delivery for dynamic adaptive streaming, IEEE INFOCOM, 2014.
[23] Raghavan P, Tompson C D. Randomized rounding: a technique for provably good algorithms and algorithmic proofs, Combinatorica, 1987.
[24] Klee V, Minty G J. How good is the simplex algorithm, WASHINGTON UNIV SEATTLE DEPT OF MATHEMATICS, 1970.
[25] Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multi-armed bandit problem, Machine learning, 2002.
[26] https://en.wikipedia.org/wiki/Knapsack_problem