Proactive Video Push for Optimizing Bandwidth Consumption in Hybrid CDN-P2P VoD Systems

Yuanxing Zhang*, Chengliang Gao*, Yangze Guo*, Kaigui Bian*, Xin Jin†, Zhi Yang*, Lingyang Song*, Jiangang Cheng‡, Hu Tuo‡ and Xiaoming Li*
*School of Electronics Engineering and Computer Science, Peking University, Beijing, China
†Johns Hopkins University, USA
‡iQIYI Co., Ltd., China
Email: *{longo, gaochengliang, guoyz, bkg, yangzhi, lingyang.song, lxm}@pku.edu.cn,
†xinjin@cs.jhu.edu, ‡{chengjiangang,tuohu}@qiyi.com

***Abstract*—Decentralizing content delivery to edge devices has become a popular solution for saving the bandwidth consumption of CDN when the CDN bandwidth is expensive. One successful realization is the hybrid CDN-P2P VoD system, where a client is allowed to request video content from a number of seeds (seed clients) in the P2P network. However, the *seed scarcity* problem may arise for a video resource when there are an insufficient number of seeds to satisfy requests to the video. To alleviate this problem, many commercial VoD systems have employed a *video push* mechanism that directly sends the recent scarce video resources to randomly-chosen seeds to serve more requests. However, the current video push mechanism fails to consider which videos will become scarce in the future, or differentiate the uploading capability of different seeds. In this paper, we propose *Proactive-Push*, a video push mechanism that lowers the bandwidth consumption of CDN by predicting future scarce videos and proactively sending them to competent seeds with strong uploading capabilities. Proactive-Push trains neural network models to correctly predict 80% of future scarce video resources, and identify over 90% of competent seeds. We evaluate Proactive-Push using a trace-driven emulation and a real-world pilot deployment over a commercial VoD system. Results show that Proactive-Push can further reduce the proportion of direct download from CDN by 21%, and save the CDN bandwidth cost at peak time by 18%.**

***Index Terms*—Video streaming, CDN, deep learning, edge computing, network traffic control**

I. Introduction

People are cutting the budget of cable TV at home, and turning to online video streaming services for "on-demand" access to TV shows and movies, a.k.a. video-on-demand (VoD) streaming services (e.g., Netflix, Hulu Plus, iQIYI, Youku). Accordingly, video service providers continuously launch new programs of various forms and introduce new strategies to provide good quality of experience (QoE) for users [1]. There are two pain points when building a VoD system. First, strategies designed for increasing client-side QoE would probably increase the distance between the source of videos and the clients and occupy much bandwidth, while massive requests to the popular videos at peak time may degrade the QoE of users due to limited bandwidth [2] or poor network condition. Moreover, the bandwidth may cost too much, as the Internet Service Provider (ISP) charges the "last-mile delivery" [3] of the system for bandwidth consumption by the overall throughput or the peak-time bandwidth metering (e.g. the 95th percentile bandwidth metering [4]). In this paper, we focus on the charging policies by the 95th percentile bandwidth metering, as it empirically costs less on the CDN.

Many solutions to reducing the bandwidth consumption of CDN have been proposed, such as redirecting requests to several appropriate edge servers by proxy server [5], caching hot videos in edge servers in advance [6], etc. Limited by the dynamic network conditions between the source of videos and the clients, server side solutions may not always perform as well as they are expected. Therefore, one prevailing way in many Asia countries for addressing this problem is to decentralize the transmission to edge devices to both reduce the bandwidth consumption of the CDN and shorten the distance between the source of videos and the clients [7]. Many types of terminal devices (clients) could be involved under this edge computing paradigm, such as smart-routers, PCs, mobile devices, etc. Resources can be transmitted among edge devices, and the peer-to-peer (P2P) technology could be leveraged to reduce the bandwidth consumption on CDN servers, without changing the CDN server strategies. In such an edge-assisted VoD system, a seed (or a seed device) may own the copies of videos needed by other clients, and a client can request the video content from the seed, instead of the direct download from the CDN. This kind of edge-assisted VoD system can be named as *hybrid CDN-P2P network*.

However, the *seed scarcity* problem may arise when there are an insufficient number of seeds available to a certain video in the P2P network for satisfying the massive P2P requests to it [8]. To alleviate the seed scarcity, commercial hybrid CDN-P2P VoD systems such as iQIYI[1] have deployed a *video push* mechanism [9]. The system directly sends the recent scarce video resources to edge devices to serve more requests to the videos through the P2P network. The current video push mechanism fails to fully address the seed scarcity problem at peak time, as it has no idea about which videos will become scarce, or which seeds will be online/competent in the future.

In this paper, we propose *Proactive-Push*, a video push

[1]http://www.iqiyi.com/

mechanism for the hybrid CDN-P2P VoD system to schedule sending the predicted scarce videos into the P2P network before the peak time starts. Meanwhile, Proactive-Push excludes the poor-quality seeds that may disappear or have low uploading bandwidth from the P2P streaming. This proactively optimizes the assignment of scarce video resources to competent seeds before the peak time, and greatly relieves the stress of the CDN caused by direct download of scarce video resources.

We evaluate the performance of Proactive-Push using the traces of real video sessions collected from 1st March 2017 to 30 June 2017, over iQIYI, one of the largest video service providers in China. iQIYI has attracted around 550 million mobile devices and more than 250 million PC users monthly, and it receives more than 6 billion hours of video-viewing time every month in 2017. Then, we deploy it over one of CDN servers that has hundreds of thousand videos and about millions of users during the time period from 1st July 2017 to 14th July 2017. Results show that under various scenarios, Proactive-Push can predict scarce videos at a precision rate of about 80%, and correctly identify 90% of competent seeds with strong uploading capabilities. Besides, Proactive-Push can further reduce the proportion of direct download from CDN by 21%, and save the last mile delivery cost by 18% in the real-world deployment.

In summary, our contributions are of threefold:

- This is the first work that establishes a deep learning (DL) based model to jointly predict the seed scarcity of videos and clients' uploading capability in the hybrid CDN-P2P VoD system.
- By the DL-based model, we propose a proactive video push mechanism that optimizes the recommendation of scarce video resources to clients with strong uploading capabilities.
- Results of trace-driven emulation and real-world deployment show that Proactive-Push has a high precision for predictions of seed scarcity of videos and clients' uploading capability, and it can greatly relieve the seed scarcity, and remarkably save the CDN bandwidth consumption in the hybrid CDN-P2P VoD system.

II. Related work

Deployment of hybrid CDN-P2P systems. The hybrid CDN-P2P system has become the most popular architecture for VoD systems in Asia. There are also many researches on the optimization of hybrid CDN-P2P architecture, by using the P2P transmission to relieve the CDN load. Xu *et al.* [10] put forward a hybrid CDN-P2P architecture for streaming media distribution and analyze its performance by simulation. Yin *et al.* [11] design a scalable hybrid CDN-P2P system considering the reliability of CDN and the scalability of P2P. Existing work has confirmed that the hybrid CDN-P2P system can adapt to the exploding growth of internet video content economically, without generating unaffordable burden on ISPs [12].

Predicting the popularity of video content. There have been many recent researches on predicting the popularity of video content. Cha *et al.* [13] analyze the popularity distribution and the popularity variation along with time based on the Youtube data. In order to improve the accuracy on predicting the popularity of videos, Flavio proposes an algorithm by extracting relevant features [14]. Besides, Li *et al.* [15] propose a propagation-based prediction framework for predicting the video popularity in online social networks.

Time series prediction. A time series is a series of data points indexed in time order, and time series prediction is the use of a model to predict future values based on previously-observed values of the time series data. Time series prediction has been proved to be a promising way in the context of quantitative finance, seismology, meteorology, etc. The auto-correlation models compute the current value in the time series as a function of a finite number of past values along with some white noise. Auto regressive moving average model (ARMA) and Auto regressive integrated moving average model (ARIMA) [16] are the most common examples in this category. Markovian models learn a stationary distribution over a predefined or automatically deciphered state space, and Hidden markov models (HMM) [17] for the hidden states. Recently, neural networks like recurrent neural network (RNN) [18] are designed to handle the sequence dependence.

III. Video Push Problem in the Hybrid CDN-P2P VoD Streaming System

In this section, we first present a brief introduction to the original video push mechanism in iQIYI, and then we formulate the video push problem.

A. System architecture of iQIYI

The system consists of clients, the CDN, the P2P network, and a video push mechanism, as shown in Fig. 1. Note that a CODEC submodule has been embedded at both server- and client-side, to prevent violating copyright of videos.

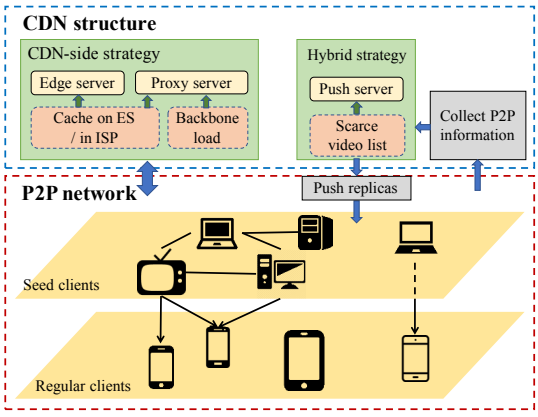

Fig. 1: The structure of a commercial hybrid CDN-P2P VoD system.

The goal of video push: Generally, a client is allowed to download video segments from the CDN or neighboring

seed clients in the P2P network. The ISP does not charge for the bandwidth consumed by video download from the P2P network to clients. Instead, it charges for the bandwidth consumed by video download from the CDN. Hence, the goal of the video push mechanism is to reduce the peak bandwidth consumed by video download from the CDN to clients. Based on our observation, the peak bandwidth consumption always happens within 21:30 to 23:00, when the total number of video requests is far greater than other time periods of a day. Therefore, we consider 21:30 to 23:00 as the peak hours of a day in this paper.

The CDN: Here, we only present the high-level structure of the system's CDN.

- *Edge server*: Clients download video resources from an edge server in the CDN. The edge server caches frequently requested video resources, i.e., scarce in the P2P network, according to the CDN cache strategy, and meanwhile it asks for dispatching from the data center if a client requests videos that are not stored on it.
- *Proxy server*: When a client requests a video, it sends DNS requests for the URL of the video to the proxy servers. The proxy servers respond to the request by some appropriate edge servers according to the proxy strategy based on the internet topology and traffic load in the backbone network.

Clients (Edge devices): There are two types of clients, and every client is required to periodically send certain control information to the P2P tracker server and the video push server. Moreover, the size of client buffer is limited.

- A *regular* client downloads the video content that the user wants to watch into its buffer, without uploading the buffer content to other neighboring clients. For example, mobile devices are regular clients that do not upload video contents into the P2P network for saving battery.
- A *seed* client, or simply *seed*, is a client that is willing to upload the video content within its buffer to neighboring clients when the video is requested by others. For example, non-mobile devices (PC, laptop) are candidate seed clients that have the capability of uploading video contents to others.

Local and remote video contents in seed buffer: The buffer of a seed client contains two parts of contents: (1) the *local* video content that the client/seed user has watched; (2) the *remote* video content that is never watched by the client/seed user himself but will be uploaded to neighboring clients.

A large-size video is usually split into multiple video segments of the same size. The segments are transmitted in the VoD system and stored in the buffer of clients. For simplicity, we simply treat each segment as a video in the following parts of this paper.

Try P2P first for download: When requesting a video, any client retrieves the list of neighboring seeds who have the video content in their buffers from the P2P tracker server, and then it tries to download the video from the P2P network as much as possible. If the video resources in the P2P network are insufficient to satisfy the requests, the client would start requesting the remaining part of the video resource from an edge server in the CDN as backup.

The current video push mechanism: The video push mechanism instructs which scarce videos will be pushed to which seed clients in the P2P network. In the current video push mechanism,

- The seed clients with free bandwidths send requests periodically to the video push server for requesting remote video contents that are scarce (recall that a video resource is *scarce* when there are an insufficient number of seeds for satisfying the requests to this video).
- The video push server maintains a list of scarce videos, which records the exact number of needed seeds for each video. Besides, the server sets up a waiting queue of client requests when multiple requests arrive simultaneously.
- The server directly pushes the most-recently scarce video resources (i.e., the videos that experience the seed scarcity problem most recently) to seeds with free bandwidth in the P2P network.

B. The video push problem

Given a video push scheme $\mathcal{A}$, a video set V and a client set U, let $C_{v,t}(\mathcal{A})$ denote the compensated number of needed replicas from CDN for video v during a short time period t, and let $P_{j,v,t}(\mathcal{A})$ represent whether the server pushes video v to client j during t.

Define a triad $(\bar{u}_{j,v,t}, p_{j,v,t}, i_{j,v,t})$ to characterize the relationship between client j and video v during time period t, where $\bar{u}_{j,v,t}$ represents the uploading capability of client j for video v, while the other two variables indicate whether client j requests playback of v and whether replica of v is in the buffer of j during time period t, respectively. Note that $u_{j,t}$ denotes the uploading capability of client j during time period t, i.e.

$$\sum_{v \in V} \bar{u}_{j,v,t} \leq u_{j,t}.$$

Therefore, the objective of the video push problem is to minimize the peak overall traffic from CDN servers, i.e.

$$\arg\min_{\mathcal{A}}\{\max_{t}\{(\sum_{j\in U}\sum_{v\in V} P_{j,v,t}(\mathcal{A}) + \sum_{v\in V} C_{v,t}(\mathcal{A}))\cdot s\}\} \quad (1)$$

s.t.

$$\sum_{j\in U} p_{j,v,t}\cdot s = \sum_{j\in U}\bar{u}_{j,v,t} + C_{v,t}(\mathcal{A})\cdot s, \quad (2)$$

$$i_{j,v,t}\cdot u_b \leq \bar{u}_{j,v,t} \leq i_{j,v,t}\cdot u_{j,t} \text{ or } \bar{u}_{j,v,t} = 0, \quad (3)$$

where s represents the size of the video segments, B is the capacity of the client buffer, u_b is the minimal rate over a transmission link to avoid the slow-loris-like effect on clients [19].

For each video, Constraint (2) prescribes that the total amount of downloaded data is equal to the amounts of

uploaded data from both the P2P network and the CDN servers. Besides, Constraint (3) imposes the restriction on the uploading links in the P2P network: the client j can only upload the data of video v during time period t if it holds the replicas in its buffer, i.e., $i_{j,v,t} = 1$; the uploading rate cannot exceed the client's uploading capability.

The video push scheme may both affect the bandwidth for pushing replicas to clients and compensating requests from clients in the future. Due to the dynamic tendency of users' requests, it is complicated to figure out the exact condition of the hybrid CDN-P2P network. Therefore, we have to carefully choose a set of features and construct an appropriate model to approximate the condition of the system.

IV. Motivation for proactive video push

A. Redundancy of the pushed video replicas at peak-time

The current video push strategy only considers pushing the recent scarce videos to the P2P network, without any forward looking over the future peak-time video scarcity that will indeed determine the CDN bandwidth cost (e.g., according to the 95th percentile bandwidth metering). Hence, it is possible that the video push server may push more video replicas to seed clients than what will be really needed at peak-time. These redundant video replicas consume bandwidth and resources that can be used for serving requests to other peak-time scarce videos. According to the statistics of the commercial system, we find that 27% of the video replicas pushed before peak-time receive no requests at peak-time, which confirms the redundancy of the pushed video replicas at peak-time.

Hence, we need a new video push strategy that can predict the video scarcity and the clients' uploading capabilities so as to avoid pushing redundant video contents.

B. Feasibility of predicting the video scarcity

The scarcity of a video is closely related to the number of requests to the video. The more requests arrive at peak-time, the more likely the video is to be scarce. According to Constraint (2), if we can foresee the number of requests to a video at peak-time, the system can calculate the number of needed seeds, and pushes an appropriate number of video replicas to the P2P network.

Three patterns in the number of video requests: We observe that there are three patterns over the number of requests to individual videos throughout a day in the commercial system, as shown in Fig. 2. Each playback trace of a video is unified by the peak-time number of video requests during the day, and then the traces in our dataset are grouped by k-means [18]. The first pattern represents the videos released usually at midnight. These videos have peak number of requests during noon and night the next day, and they attract more attention at noon. The second pattern corresponds to those videos whose peak number of requests is achieved at night, while they also exhibit a lower peak at noon. The third pattern portrays the tendencies which raise up around evening and only have one peak at night. These three conspicuous patterns indicate the feasibility of predicting the peak number of video requests.

C. Feasibility of predicting the uploading capability of clients

The uploading capability of a client is dependent on its uploading rate in the P2P network, which may vary over time. It is important to predict which clients will have a high uploading capability at peak-time, and push the predicted scarce videos to the predicted competent clients.

Three patterns in the uploading capability of clients: Then, we observe the uploading rate of individual clients, and we unify the uploading rate of every client in a day by its peak uploading rate of the day. Grouped by k-means, there are three patterns of the uploading capability, as shown in Fig. 3. The second pattern shows the greatest uploading capability in the afternoon, while the greatest uploading capability of the third pattern occurs at night. On the contrary, the first pattern shows a relatively constant curve, which implies an always-on seed client in the P2P network.

V. The Proactive-Push Mechanism

Proactive-Push consists of three components: (1) prediction of the number of requests to individual videos at peak-time; (2) prediction of uploading capabilities of individual clients; and (3) an allocation strategy that addresses the video-client assignment problem—push predicted scarce videos to predicted competent clients before the peak-time.

A. Prediction of the number of requests to individual videos

1) Features in focus: To have a better prediction of the peak number of requests to a single video, we collect as much information related to the video as possible. There are two categories of features in focus: the *contextual features* that would be variant within a short time (e.g., the number of video requests); and the *semantic features* that would not change within a short time (e.g., video tags including genres, actors, actresses, creators).

Contextual features: There are two types of contextual features that can contribute to the prediction:

- As indicated in Section IV, the number of video requests during off-peak time has a positive correlation with the peak number of the video requests at peak-time;
- The local video content in the client/seed's buffer that the seed user has watched can reflect the popularity of the video.

Semantic features: Usually, most video requests direct to a few popular videos. According to the statistics, we find that during peak-time, several genres on videos obtain an extremely high number of requests. The genres are carefully labeled by the uploaders and the copyright owners in order to make them easy to be searched or accessed by users.

To better describe the features, they are transformed into 01 coding vectors. As the genres are independent of the order when they are labeled, we are inspired by the continuous bag of words language model [20] and we are seeking to

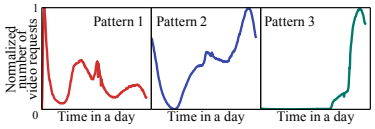

Fig. 2: Three observed patterns over the number of requests to individual videos throughout a day in the commercial system.

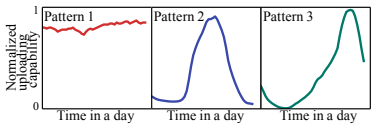

Fig. 3: Three patterns over the uploading capability of a single client throughout a day.

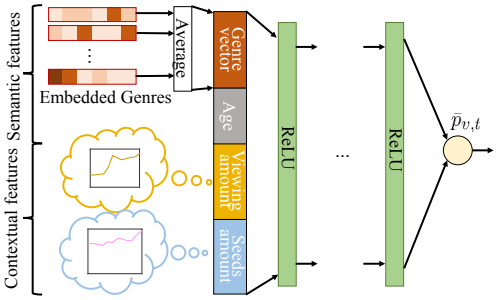

Fig. 4: The neural network on predicting the peak number of video requests of single video.

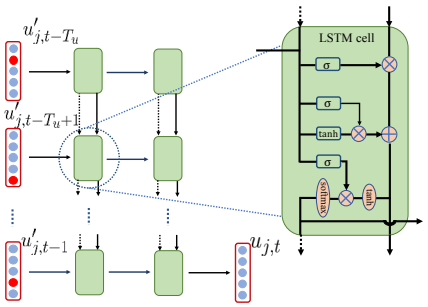

Fig. 5: The neural network on predicting the uploading capability levels of clients.

learn high dimensional embeddings for each genres in a fixed vocabulary. Specifically, we solve the singular value decomposition (SVD) [21] to learn the embeddings.

Besides, the age of a video since its release may also contribute to predicting the peak number of video requests. Based on our analysis on the data, it is revealed that most videos reach the peak number of video requests during the first couple of days since they are released online.

2) Design of the Neural Network (NN) model: It is non-trivial to figure out the complex correlation among different features. Hence, we introduce the feed-forward neural network (FNN) model [18] to learn their correlation. The neural network takes both the semantic and contextual features of a video as inputs, and it provides the prediction of the peak number of video requests as outputs. The structure of the model is illustrated in Fig. 4.

Input of NN: As there may be many tags of genres on each video, we should compute the mean of embeddings of the related genres. Accordingly, a descriptive vector with a fixed length is obtained. Meanwhile, the contextual information is appended to the vector. The vector of video v during time period t, denoted as $\mathbf{x_{j,t}}$, is then set as the input to the neural network.

Output of NN: The goal is to learn a set of functions f_t for every time period t, which can predict the number of video requests during peak hours, namely $\bar{p}_{v,t_p}$.

Loss function: As the model may overfit on the training set, we should add regulation to the loss function. Let $\bar{p}'_{v,t_p}$ denote the real number of requests to video v during peak hours. Therefore, the loss function can be written as:

$$L_t(V;\theta) = -\frac{1}{|V|}\sum_{v\in V}|\bar{p}'_{v,t_p} - \bar{p}_{v,t_p}| + \lambda||\theta||^2, \qquad (4)$$

where λ is the regulation coefficient.

Activation function: Aiming at improving the precision of the neural network and accelerating the training speed, the hidden layers should be elaborately designed. Rectified linear units (ReLU) are used as the activation function for improving the network training speed and the classification accuracy [18].

B. Prediction of the level of the uploading capability of clients

As previously observed from the three patterns in Section IV, there is no positive correlation between the uploading rate of a client and its peak value, and we then seek to predict the pattern of the uploading capability of clients. Here, we define five levels of the uploading capability of clients to represent the possible patterns.

Levels of the uploading capability of clients: We use the average uploading rate reported by the clients as an indicator to determine the client's uploading capability $u_{j,t}$. According to distribution of clients' uploading rate and the prevailing bandwidth rate provided by ISPs, we divide the uploading rate into five levels, from 0 to 4, where level 0 represents that the client is currently offline or cannot connect to other peer clients in the P2P network; and clients in level h' have higher average uploading rate than clients in level h'', if $h' > h''$.

LSTM for prediction of uploading capability levels: As the client can periodically upload its uploading rate to the server, the prediction could be transformed to a time series prediction problem. The RNN model has been proved to be efficient in solving the time series prediction problem. We then implement an anti-long-term-dependency version, i.e., Long Short-Term Memory (LSTM) [22] network, to predict the level of the uploading capability of clients. An LSTM is well-suited to learn from the past experience to predict time series, given time lags of unknown size and bound between important events. The structure is shown in Fig. 5, where the input for each LSTM cell is an one-hot encoding vector $\mathbf{u}_{j,t}$, representing the uploading capability level of j at time period t.

C. Pushing scarce videos to competent clients

1) Setting the pushing target: Let $\mathbb{U}_t$ denote the distribution of uploading capability of clients in the P2P network at time period t. According to Constraint (2), the video push server would try to place appropriate video replicas to seed clients to decrease the bandwidth consumption from CDN. Given a situation that a number of ι clients are cooperatively uploading data for one video request, then the probability of no bandwidth consumption from CDN could be written as

$$P_t(\sum_{\xi\in[1,\iota]} u_\xi \geq s \big| u_\xi \sim \mathbb{U}_t).$$

The system would set a target probability threshold ϕ, in expectation that each request would be fully fulfilled with possibility ϕ, i.e.,

$$P_t(\sum_{\xi\in[1,\iota]} u_\xi \geq s \big| u_\xi \sim \mathbb{U}_t) = \phi.$$

Therefore, we could obtain a simplified pushing target to promise that every request could be responded by at least a number of ι seed clients. That is,

$$\forall v \in V, \sum_{j\in U} i_{j,v,t} \geq \iota \sum_{j\in U} p_{j,v,t}. \tag{5}$$

2) Maximizing the P2P throughput: We need to address the video-client assignment problem that recommends scarce videos to an appropriate client, so that the bandwidth consumption from CDN could be minimized.

Let $\eta_{v,t}$ denote the required number of replicas to be pushed into the P2P network for video v at time period t. According to Equ. (5), we have

$$\eta_{v,t} = \max\{\iota \sum_{j\in U} p_{j,v,t} - \sum_{j\in U} i_{j,v,t}, 0\}.$$

Given two clients j_1 and j_2, with $u_{j_1,t}$ greater than $u_{j_2,t}$. Then j_1 should undertake the uploading of videos that require more replicas, as it has a greater capability to upload more data to other clients in the P2P network. Otherwise, the uploaded data of j_1 could be limited as the P2P transmission could have already been saturated, while j_2 may contribute little to the system due to its limited uploading capability. Therefore, the server should place scarce videos that require more replicas to the clients with a greater uploading capability.

The procedure of Proactive-Push: We present the step-by-step procedure of Proactive-Push in Algorithm 1, where $\mathcal{Z}$ represents the time period for generating the list of to-be-pushed scarce videos based on the prediction model in Sec. V-A. Every time when the server receives a request from client, a report of its uploading rate is also included, and the server would then determine the future uploading capability level of the client by the model in Sec. V-B.

The modified structure of the video push server is shown in Fig. 6, which maintains a scarce video list recording the videos with seeds in shortage, a seed client queue that sorts the clients by their uploading capability levels, and a pushing limitation controller that adjusts how much video content will be pushed in the future, based on the estimation of the total number of video requests at peak-time in the system (that will be introduced next).

Algorithm 1 Procedure for Proactive-Push.

for every time period with length of $\mathcal{Z}$ **do**
  ▷ Generate a list of scarce videos to be pushed $\mathcal{L}$ according to the prediction model.
  Estimate the total number of video requests as the pushing limitation $\widetilde{\Omega}$.
  ▷ Set the seed client queue ω as empty.
  for every time window T **do**
    repeat
      ▷ Receive client j's request.
      ▷ Add client j into ω.
    until Time window ends.
    ▷ Sort clients in ω according to their uploading capability levels, and remove those with level 0.
    ▷ Extract the top-$\mathcal{K}$ scarce videos from $\mathcal{L}$ to create a temporary list $\mathcal{L}'$ based on $|\omega|$, $|\mathcal{L}|$ and $\widetilde{\Omega}$.
    for every client j in ω in order **do**
      ▷ Push the top-$\mathcal{S}$ different videos to client j.
      ▷ Update $\mathcal{L}'$.
    end for
    ▷ Merge $\mathcal{L}'$ back to $\mathcal{L}$.
    ▷ Sort $\mathcal{L}$ by η_v, the required number of replicas to be pushed into the P2P network for video v.
  end for
end for

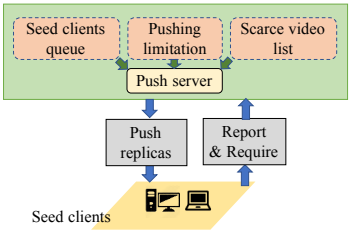

Fig. 6: The structure of the push server with Proactive-Push.

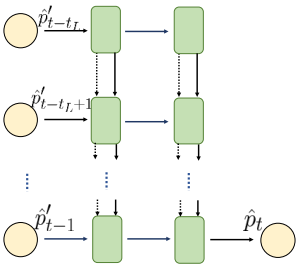

Fig. 7: The neural network on predicting the total number of video requests at peak-time.

D. Estimation of the total number of video requests at peak-time

Recall that we focus on the policy charging the VoD service provider by the peak-time CDN bandwidth consumption (or 95th percentile), which is dependent on, N_c, the number of video requests directly sent to the CDN. Let $\hat{p}_{t_p}$ denote the total number of video requests at peak-time, let ρ denote the proportion of video requests that can be satisfied by downloading from seeds in the P2P network, and thus $N_c = \hat{p}_{t_p}(1-\rho)$. The value of ρ can be easily obtained by having an averaged value over the statistics during the

previous couple of days. Next, we introduce how to estimate $\hat{p}_{t_p}$ using the LSTM model.

By the summation of the number of requests to all individual videos, we have

$$\hat{p}_{t_p} = \sum_{v \in V} \bar{p}_{v,t_p} \tag{6}$$

We periodically collect the total number of video requests in the system, and convert the prediction problem to a time series problem, which we can use LSTM again to solve rather than using FNN. The structure of the LSTM in this case, as shown in Fig. 7, is similar to the one used in predicting the uploading capability levels of clients.

Let $\hat{p}_t$ denote estimated total number of video requests during time period t. Let $\hat{p}'_t$ denote real total number of video requests during time period t, which can be obtained from the dataset. Then $\hat{p}_t$ can be estimated through the LSTM network with the real total numbers of video requests during the previous time periods $(t - t_L)$, ..., $(t - 1)$, i.e.,

$$\hat{p}_t = \text{LSTM}(\hat{p}'_{t-t_L}, \ldots, \hat{p}'_{t-1}; \theta_{LSTM}), \tag{7}$$

where t_L is the number of previous time periods used for the estimation. We could put the estimated value $\hat{p}_t$ back to the tail of the time series as the input of the LSTM network, and then $\hat{p}_{t+1}$ can be calculated. By parity of reasoning, we could predict the total number of video requests for the following days. The model takes the total number of video requests everyday in the previous three months as training data, the length of which is denoted by T_l.

VI. Evaluation

Datasets. We collect the proprietary data on servers of the commercial system from 1st March, 2017 to 30th June, 2017. The data records the snapshot of the number of video requests, and replicas in the P2P network of a set of videos every five minutes. Besides, the video requests from the clients and their uploading and downloading rates are also recorded. About 2,000,000 videos are involved, and we have access to the semantic and contextual information of them.

Implementation. We use a 6-hidden-layers FNN with 250 hidden nodes in each layer to predict the number of requests to each video during peak-time. Besides, we set half an hour as a time period, i.e., 48 time periods per day, and we use the uploading rate of a client in the previous 48 time periods to predict the uploading capability level in the next time period. Similarly, the total number of video requests is predicted based on the data in previous 48 time periods. The LSTM models are both set to have two layers. We find that using more hidden nodes or more layers will not further improve the performance of the models. We use the 10-fold cross-validation [18] for training the models.

Proactive-Push is deployed over the video push server of the commercial system. We use the data from 2nd June, 2017 to 29th June, 2017 to train the FNN model, which does not contain any national holidays that may cause interference to our observations. To predict the daily pattern, every half hour we set up a prediction model based on the training data collected in the previous hour. There are about 200 thousand records for each model, and in total we need to set up 48 models for a daily prediction model (one model every half hour, and 24 hours a day).

We modify the ways of generating the scarce video list and responding to the video request from clients. For every half hour, the server collects the system statistics of the previous hour, predicts the seed scarcity of videos, and decides the number of needed seeds. The server ranks all videos that suffer from the seed scarcity problem, and prioritizes videos by the number of needed seeds. The server responds to a client's request by recommending the top videos in the scarce video list, as described in Section V-C. The format of the scarce video list is consistent with the existing format to keep the backward compatibility to the existing system.

The network flow model. Usually, the amount of data through P2P streaming flow between a seed and a client is not recorded in any real-world system due to the extremely high cost of doing so. The aggregate rate of uploading/downloading to and from every client is known, but the exact rate of the flow on each link between two clients is unknown. Hence, we refer to the network flow model [23] to emulate the P2P streaming over links between a seed and a client, and the calculation of proportion of P2P streaming can be transformed to a max-flow problem.

A. Precision of predicting the peak-time number of requests to individual videos

The accuracy of the prediction on the peak-time number of requests to individual videos would determine whether the appropriate number of replicas should be pushed to the P2P network. We use the data of one month length to train the FNN model and test over the next one-week data. In other words, the experiments run throughout the dataset, and the parameters get updated by every week of data. Besides, we only focus the videos with more than 40 concurrent requests in this experiment.

We first examine the appropriate length of time period for observing the contextual features that will be put into the FNN network. The length is set as 0.5 hour, 1 hour, 1.5 hours, or 2 hours, and the precisions are plotted in Fig. 8. Apparently, the prediction with 0.5 hour data may miss some critical information and therefore performs a lower precision. Meanwhile, when the length becomes more than one hour, then the precision would not get an evident increment. Hence, we fix the length of time period for observing contextual features as one hour for saving the storage and accelerating the computation.

Then we inspect the precision of the FNN model with semantic and contextual features. We compare it to the supported vector regression (SVR) and linear regression (LR) models with the same input data, while the FNN with only contextual features is also compared. The models are tuned by 10 fold cross validation on the training set to reach good performances. Besides, the original push strategy serves as

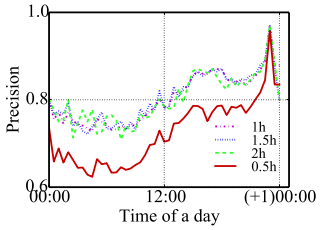

Fig. 8: The impact of different lengths of data on predicting the peak-time number of requests to individual videos.

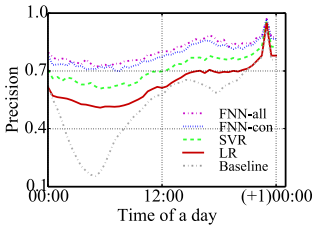

Fig. 9: The precision on predicting the peak-time number of requests to individual videos, under different prediction models.

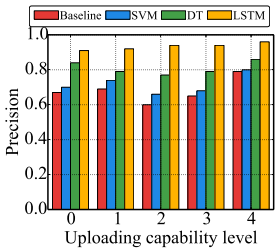

Fig. 10: The precision on predicting the uploading capability level of clients.

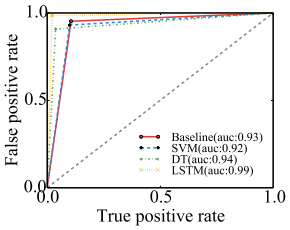

Fig. 11: The ROC on predicting the uploading capability level of clients.

baseline. Results in Fig. 9 show that our FNN model in Proactive-Push outperforms other methods, and the use of semantic features does help contributing to the prediction on the peak-time number of requests to individual videos.

B. Precision of predicting the uploading capability level of clients

Similarly, we use the data of one month length to train the LSTM model and test on the next one-week data. Here, we focus on the precision of correctly predicting the uploading capability level, and compare the LSTM model with supported vector machine (SVM) and Decision Tree (DT). We use the latest reported uploading rate as the baseline approach. As illustrated in Fig. 10, the LSTM model increases the overall precision by about 27% compared with the baseline strategy. Besides, the LSTM model produces a similar and high precision on all five levels of the uploading capability. The ROC curve is shown in Fig. 11, which implies that the LSTM model could provide an accurate prediction on the uploading capability of clients.

C. Error rate in estimating the total number of video requests at peak-time

Regarding the estimation of the total number of video requests at peak-time, we compare four models: the LSTM model using the data from the past few hours, the LSTM model using the peak value of the previous days, LR and ARMA using the data from the past few hours. Fig. 12 exposes error rate in the estimation, where a positive (or negative) error rate implies that the algorithm estimates more (or less) than the real total number of video requests at peak-time. Results indicate that using total number of video requests at peak-time of previous days could generating the highest error rate, while our LSTM model using data from previous hours outperforms others, with a high precision of over 97%.

D. Saving CDN bandwidth consumption

We collect the data from 2nd, June, 2017 to 29th, June, 2017 to observe the performance of three video push mechanisms in saving the CDN bandwidth consumption, namely the original video push mechanism, Proactive-Push, and the oracle or ideal strategy (which generates the scarce video list based on the ground-truth of the P2P network information and the uploading capability of every client during peak-time). The pushing target ι is set as 20, same as the value used in the commercial system.

We monitor the proportion of direct download from CDN edge servers of the 95th percentile of every day, and plot the histogram by week in Fig. 13. Our proposed Proactive-Push mechanism can further reduce the proportion download from CDN by 21%, which makes a great step improvement towards the oracle push strategy.

Aiming at examining the real performance of our proposed strategy, we implement a pilot deployment on one of the trackers in the real commercial VoD system. Meanwhile, another tracker implementing the original push strategy is taken as comparison. The experiments were executed from 1st July, 2017 to 14th July, 2017, and the CDN bandwidth cost is shown in Fig. 14. Note that the cost (i.e., the values of the y-axis in the figure) is scaled for hiding the real value and protecting the privacy of the commercial system. On average, the proposed strategy reduces the CDN bandwidth cost by 18% compared with the original video push strategy.

E. Setting the pushing target parameter ι

In our experiments and the real-world deployment, the pushing target $\iota = 20$. Here, we test the impact of this pushing target parameter. We demonstrate the proportion of direct download from CDN in Fig. 15, varying ι from 1 to 30 using the data from 2nd, June, 2017 to 29th, June, 2017. It is clear that the proportion of direct download from CDN becomes stable when $\iota = 20$, but it rises sharply when ι is greater or smaller.

VII. Conclusion

As more and more people prefer watching videos through Internet, the last-mile delivery cost for transmitting video segments from CDN servers to clients keeps increasing. One prevailing solution for cutting the expenses on CDN in Asia is to offload the transmission to edge devices and implement a hybrid CDN-P2P VoD system, where a video push mechanism

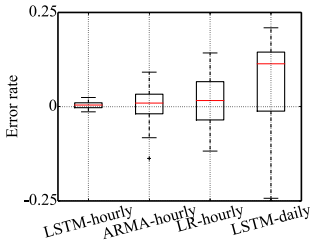

Fig. 12: The error rate on estimating the total number of video requests in the system.

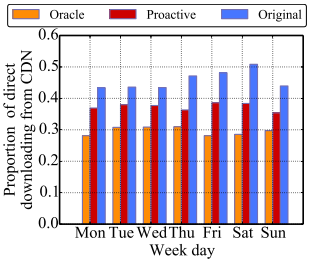

Fig. 13: The proportion of direct download from CDN.

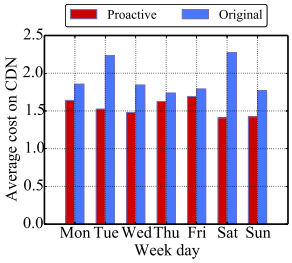

Fig. 14: The average CDN bandwidth cost.

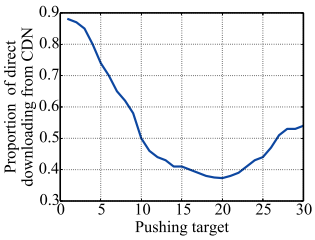

Fig. 15: The sensitivity of pushing target ι.

is proposed by leveraging the free bandwidth during off-peak hours and the vacant clients. In this paper, we analyze the patterns of the number of video requests and the uploading rate of clients in a day according to traces of a real commercial VoD system. Then we utilize neural network models to predict the number of requests of individual videos during peak hours, the uploading capability level of clients, and the total number of video requests during peak hours. Based on the three prediction models, we propose a novel video push mechanism named Proactive-push. Emulations on real traces show that our proposed strategy can further reduce 21% of the proportion of direct download from CDN compared with the original video push strategy. Our pilot deployment over iQIYI shows that Proactive-push can save 18% more cost on the CDN servers at peak time.

ACKNOWLEDGMENT

This work is partially supported by the National Key Research and Development Program No. 2017YFB0803302, the National 973 Grant No. 2014CB340405, and the National Natural Science Foundation of China under Grant Nos. 61572051, 61625101 and 61632017.

REFERENCES

[1] Junchen Jiang, Shijie Sun, Vyas Sekar, and Hui Zhang, "Pytheas: Enabling data-driven quality of experience optimization using group-based exploration-exploitation.," in *NSDI*, 2017, pp. 393–406.

[2] Oliver Hohlfeld, Enric Pujol, Florin Ciucu, Anja Feldmann, and Paul Barford, "A qoe perspective on sizing network buffers," in *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 2014, pp. 333–346.

[3] Hao Yin, Xuening Liu, Geyong Min, and Chuang Lin, "Content delivery networks: a bridge between emerging applications and future ip networks," *IEEE Network*, vol. 24, no. 4, 2010.

[4] Rade Stanojevic, Nikolaos Laoutaris, and Pablo Rodriguez, "On economic heavy hitters: shapley value analysis of 95th-percentile pricing," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 75–80.

[5] Vijay Kumar Adhikari, Yang Guo, Fang Hao, Matteo Varvello, Volker Hilt, Moritz Steiner, and Zhi-Li Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1620–1628.

[6] Sem Borst, Varun Gupta, and Anwar Walid, "Distributed caching algorithms for content distribution networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[7] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[8] Hai Jiang, Jun Li, Zhongcheng Li, and Xiangyu Bai, "Efficient large-scale content distribution with combination of cdn and p2p networks," *International Journal of Hybrid Information Technology*, vol. 2, no. 2, pp. 4, 2009.

[9] Ming Ma, Zhi Wang, Ke Su, and Lifeng Sun, "Understanding the smartrouter-based peer cdn for video streaming," *arXiv preprint arXiv:1605.07704*, 2016.

[10] Dongyan Xu, Sunil Suresh Kulkarni, Catherine Rosenberg, and Heung-Keung Chai, "Analysis of a cdnp2p hybrid architecture for cost-effective streaming media distribution," *Multimedia Systems*, vol. 11, no. 4, pp. 383399, 2006.

[11] Hao Yin, Xuening Liu, Tongyu Zhan, Vyas Sekar, Feng Qiu, Chuang Lin, Hui Zhang, and Bo Li, "Design and deployment of a hybrid cdn-p2p system for live video streaming: experiences with livesky," in *Proceedings of the 17th ACM international conference on Multimedia*. ACM, 2009, p. 2534.

[12] Cheng Huang, Angela Wang, Jin Li, and Keith W Ross, "Understanding hybrid cdn-p2p: why limelight needs its own red swoosh," in *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2008, p. 7580.

[13] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 5, pp. 13571370, 2009.

[14] Flavio Figueiredo, "On the prediction of popularity of trends and hits for user generated videos," in *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 2013, pp. 741–746.

[15] Haitao Li, Xiaoqiang Ma, Feng Wang, Jiangchuan Liu, and Ke Xu, "On popularity prediction of videos shared in online social networks," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013, p. 169178.

[16] Terence C Mills, *Time series techniques for economists*, Cambridge University Press, 1991.

[17] Yi Sun, Xiaoqi Yin, Junchen Jiang, Vyas Sekar, Fuyuan Lin, Nanshu Wang, Tao Liu, and Bruno Sinopoli, "Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*. ACM, 2016, pp. 272–285.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, http://www.deeplearningbook.org.

[19] Saman Taghavi Zargar, James Joshi, and David Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," *IEEE communications surveys & tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[21] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al., *Introduction to information retrieval*, vol. 1, Cambridge university press Cambridge, 2008.

[22] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[23] Andrew V Goldberg, Éva Tardos, and Robert E Tarjan, "Network flow algorithms," Tech. Rep., DTIC Document, 1989.